0. Introduction

FORTRAN 77 software is provided to assist users in unpacking and using some of the available binary data products. As discussed in each product description, it is assumed that the user has the low-level and generally machine-dependent capabilities of 1) transferring a binary block into memory and 2) then extracting into INTEGER variables the bit strings whose lengths are specified. The two capabilities are discussed briefly in secs. 1-3, together with the efficiency and machine-portability considerations that have constrained the design of product formats. A more general discussion including the advantage in execution time and storage relative to traditional techniques can be found in [3].

Source code listings for the available software appear under the filenames given in Table H0-1. Files are listed on pp. H6-H46 (except that the information in LLN2F1 appears on the 2° box map in supp. G). In addition, the files can be furnished by NCAR's Data Support Section in machine-readable form.

Table H0-1
Available User Software

Filename	Level	Purpose
BOXLIB	.01J	tools for working with 2°, 4°*, and 10° boxes, or Marsden Squares
QI9	.01G	read and print MSU.2
QI12	.01D	read and print CMR.4
QI21	.01D	read and print MSUG.1 group 1
QI22	.01D	read and print MSUG.1 group 2
QI24	.01C	read and print DSU.2
QL14	.01C	read and print MST.3
QL16	.01C	read and print TRP.1
QL21	.01C	read and print CMR.5
QL28	.01C	read and print MSTG.1 group 3
QL29	.01C	read and print MSTG.1 group 4
QL30	.01C	read and print MSTG.1 group 5
QL31	.01C	read and print MSTG.1 group 6
QL32	.01C	read and print MSTG.1 group 7
RDINV	.01B	read and print INV.3
READER	.01B	read landlocked file LLN2F1
LLN2F1	n/a	landlocked file

* 4° boxes are similar to 2° boxes. BOX4-1 and -4052 are dedicated to the exact North and South poles, respectively; the remaining boxes 2 through 4051 each enclose four 2° boxes (number 2 has BOX2-2, -3, -182, -183; number 3 has BOX2-4, -5, -184, -185; etc.).

Software may require some modification to work properly on a given machine, because of differences in FORTRAN and computer characteristics, or if the machine dependent capabilities discussed in secs. 1-3 are not available or differ in their implementation. Table H0-2 summarizes known, potential incompatibilities for each filename.

Table H0-2
Potential Incompatibilities

Incompatibility	Filename					Reference
	BOXLIB	QI9-QL32	RDINV	READER	LLN2F1	
FORTRAN 66	X	X[a]	X		n/a	--
7-char variables		X[b]			n/a	--
DATE	X	X	X		n/a	this section
TIME	X	X	X		n/a	this section
BPW (bits/word)		X	X[c]		n/a	this section
BUFFER IN		X	X		n/a	sec. 1
UNIT		X	X		n/a	sec. 1
LENGTH		X			n/a	sec. 1
RPTIN		X[d]			n/a	sec. 1
GBYTES		X	X		n/a	sec. 2
DEC computer		X	X		n/a[e]	sec. 3

[a] Limited to use of the PARAMETER statement, those parameters in the DIMENSION and DATA statement, and the apostrophe to delimit literals in PRINT and FORMAT statements.
[b] Only one, INDEXCK.
[c] Called WRDSIZ.
[d] Referenced but never called in the default implementation (since RPTOFF = 1, BUFFER IN is called instead). On systems that are rigorous in satisfying program externals, this reference should be made into a comment.
[e] Unless input as binary data.

The more minor of these incompatibilities are discussed in the following; see the referenced section for information about others.

- DATE
  This subroutine returns " yy/mm/dd." as type CHARACTER*10.

- TIME
  This subroutine returns " hh.mm.ss." as type CHARACTER*10.

- BPW
  The INTEGER bits per word is set by default to 60, and must be changed to match the machine word size.

1. Binary Input

The method of handling binary input depends on two levels of organization that are commonly used in storing data on magnetic tape and disk. First, a *logical* record is the amount of data a user desires access to in one input operation. Examples are an individual monthly summary (for MSTG 384 bits long), or an individual report (for CMR.5 192 bits long). Second, a

block (or *physical* record) is the amount of data a user may be required to access in one input operation because of hardware or system software limitations, and which is characterized by system-recognizable boundaries of various sorts between blocks. Usually, shorter logical records are blocked together into larger physical records for efficiency of storage and input/output (i/o). Although a block may be the real unit of input, in many cases system software can make this distinction transparent to the user.

The software provided here makes use of a non-ANSI but relatively common feature called BUFFER IN to input a binary block, sometimes concurrently with the calling program. The form of BUFFER IN used is

```
BUFFER IN(LUN,M) (K(1),K(N))
```

where LUN is the unit designator, K is an array that will receive the block, N is at least the number of words required to hold a block and no more (on some machines less) than the DIMENSION of K*, and M is a machine-dependent parameter for input mode. The function UNIT must be checked before K is used, to be sure BUFFER IN is done

```
JEOF=UNIT(LUN)
```

and JEOF can be

−1 if ready,
0 if end-of-file,
+1 if parity error.

The UNIT check must be delayed as long as possible to allow BUFFER IN to work concurrently with the intervening statements. This was not possible in these programs because only one buffer was used; in order to improve clock performance a "ping-pong" approach that switches between two buffers could be used. Once UNIT has been checked, the LENGTH(LUN) function can be used. It will return the number of words transferred into K.

Block sizes have been chosen that are evenly divisible by 64-bit and 60-bit words, and thus also by any smaller word size that divides evenly into 64 or 60 (e.g., 16, 32). This is convenient for BUFFER IN, as well as for alternative techniques. One alternative is to read a block in "A*n*" format where *n* is the number of characters per word. For example, on a 32-bit IBM machine with 8-bit characters,

```
      INTEGER K(1800)
100   READ(1,200) K
200   FORMAT(1800A4)
```

will read one 57,600-bit block (MSTG).

---

* Programs QI9-QL32 have this dimension set to the integer parameter DIM BUF = (1006 * 64−1)/BPW + 1 for compatibility with RPTIN. Since RPTIN is not called in the default implementation, DIM BUF can be reduced, if necessary, to the length required to hold one full block (plus 6 initial control words).

Logical record sizes have also been chosen that are evenly divisible by 64-bit words. This increases the likelihood, on a given machine, that it will be possible to read one logical record at a time. On a 60-bit CDC machine with 6-bit characters,

```
      INTEGER K(4)
100   READ(1,200) K
200   FORMAT(3A10,A2)
```

will read one 192-bit logical record (CMR.5), provided a "record manager" available with the operating system is advised by

```
FILE(TAPE1,RT = F,FL = 32,RB = 150)
```

to supply a 32-character logical record blocked 150 for every READ.

Binary input can be further simplified on machines where the RPTIN utility is available, and where the data are in RPTIN format. This utility was developed at NCAR for unblocking variable-length logical records, such as LMR, but will work equally well on fixed-length records. A complete description of RPTIN including some of its additional features can be found in [3]. In case it is available, RPTIN is offered as an option in this software, which requires that the RPTOFF parameter be changed from its default setting of 1 (indicating that RPTIN is off) to 0 (indicating that RPTIN is on). Otherwise, RPTIN will be an "unsatisfied external" that will never be called.

2. Bit-String Manipulations

After a binary block or record is transferred into memory, it will be necessary to extract into INTEGER variables the desired bit strings whose lengths are specified. Subroutines GBYTES and GBYTE are available on some machines for this purpose (together with reverse capabilities SBYTES and SBYTE as described in [3]). GBYTES is used to move N strings of constant-length-B bits from packed array P to unpacked array U, after initially skipping Q bits, and skipping S bits between each string. The call is

```
CALL GBYTES(P,U,Q,B,S,N)
```

where

P and U are indeterminate type arrays of sufficient size,
Q,B,S, and N are integers,
$1 \leqslant Q <$ word size, and $1 \leqslant B \leqslant$ word size.

If only one string is required,

```
CALL GBYTE(P,U,Q,B)
```

should be used. In improved implementations the restriction that Q be less than word size is dropped, easing code portability.

Where GBYTES and GBYTE are not available or where efficiency is the primary consideration, other techniques can be used. The Boolean operations AND, OR, SHIFT, and MASK are available on some machines; if not, it is possible to simulate them using integer arithmetic. In many cases string lengths have been chosen that are multiples of 8 bits, in which case it may be possible to treat them as characters on some machines.

3. Note for Users on DEC Equipment

All COADS packed-binary formats were designed and documented using the convention of numbering bits from high-order to low-order within words, and words are thought of as going from lowest address to highest address. This is convenient since it results in simple left to right representation of the data in a string of bits. Most large computers use this convention (IBM, Control Data, Cray, etc.) and most packed-binary formats have been designed using this convention. When 9-track tapes are read or written on such systems, the first 8-bit byte is accessed from or stored in the high-order 8 bits of the first word in the memory i/o buffer. Succeeding bytes are stored in the next lower 8 bits until the first word is filled, and storing continues in the high-order bits of the second word of the buffer.

Since DEC uses a low-order to high-order convention on bits and words, the interpretation of formats using the COADS convention can be somewhat confusing. When 8-bit bytes are read from a 9-track tape on DEC equipment, the first byte on the tape goes into the low-order 8 bits of the first word in the input memory i/o buffer. The result of this is that the 8-bit bytes within each DEC word are in reverse order of what is intended in the format. For example, if the format specifies that the first 12 bits of a data record represent a data value, after a tape is read on a DEC system these 12 bits are contained in the low-order byte followed by the high-order 4 bits of the next higher order byte.

This problem has been solved in different ways by various DEC installations. NCAR has a special version of GBYTES written for local DEC equipment. This routine allows users to think of the data as a string of bits in the COADS sense and access various sized strings of bits in the proper order. A listing of the routine may be requested from NCAR's Data Support Section.

```
      PROGRAM TEST                                                          00110
      CHARACTER*10 LEVEL*6,DTE,TME                                          00120
      INTEGER UNIT                                                          00130
      DATA LEVEL/'.01J. '/                                                  00140
      CALL DATE(DTE)                                                        00150
      CALL TIME(TME)                                                        00160
      PRINT 1,LEVEL,DTE,TME                                                 00170
    1 FORMAT('1BXPORT',3A)                                                  00180
      WRITE(UNIT,1) LEVEL,DTE,TME                                           00190
      RETURN                                                                00200
      END                                                                   00210
C     ***************************************************************     00220
C     -----BXPORT, SOURCE CODE FOR BOXLIB                                   00230
C          A LIBRARY OF TOOLS FOR USING BOXES AND OTHER GLOBAL              00240
C          GRID SYSTEMS, E.G. MARSDEN SQUARES. THE BOX SYSTEMS ARE:         00250
C            GENERIC NAME    SPECIFIC NAME    POLAR BOXES    X-ORIGIN      00260
C            ============    =============    ===========    ========      00270
C                BOX2            BX16202           YES          0E         00280
C                BOX4            BX4052            YES          0E         00290
C                BOX10           BX648              NO          30E        00300
C                                                                           00310
C     ===1========2========3========4========5========6========7==         00320
C                                                                           00330
C     -----------REVISION HISTORY-----------------------------------       00340
C      LEVEL AUTHOR DATE       DESCRIPTION                                  00350
C      ===== ====== ========== ====================                         00360
C      .01A. ---    83/07/20.  ORIGINAL VERSION TAKEN QLIBS.01I VIA F45     00370
C      .01B. SDW    83/07/21.  UPDATES BOX10 TOOLS TO CURRENT SYSTEM        00380
C      .01C. SDW    84/05/02.  FIX ERROR IN <XYBQ>, COMMENT OUT <XYMSQ>,    00390
C                              AND ADD <B1026>.                             00400
C      .01D. TSP    84/10/05.  FIXED <B10XYO> TO ADJUST FOR 30 DEGREE       00410
C                              SHIFT OF B10 SYSTEM                          00420
C      .01E. TSP    84/10/08.  FIXED ERRORS IN <MSQB10>                     00430
C      .01F. TSP    84/10/08.  FIXED <XYMSQ> AND <MSQXYO>                   00440
C      .01G. TSP    84/10/09.  DELETED <B25> AND <B52>, TRIMMED ALL         00450
C                              LINES TO 72 CHARACTERS MAXIMUM               00460
C      .01H. TSP    84/10/09.  DELETED <B5XYO>, <MSQ5>, AND <XYB5>          00470
C      .01I. TSP    84/10/10.  CHANGED NAMES OF SOURCE AND                  00480
C                              OBJECT CODE.                                 00490
C      .01J. TSP    84/10/15.  DELETED BOX5 AND AUTHOR COMMENT LINES.       00500
C     ---------------------------------------------------------------       00510
C     ===1========2========3========4========5========6========7==         00520
      INTEGER FUNCTION B10MSQ(MSQ)                                          00530
C     -----------EQUALS -1 IF ILLEGAL MSQ ELSE EQUALS EQUIVALENT B10       00540
      IMPLICIT INTEGER(A-Z)                                                 00550
      IF(MSQ.GE.1.AND.MSQ.LE.288)THEN                                       00560
        SQR=MSQ+35                                                          00570
      ELSE IF(MSQ.GE.300.AND.MSQ.LE.623)THEN                                00580
        SQR=-1*(MSQ-300)                                                    00590
      ELSE IF(MSQ.GE.901.AND.MSQ.LE.936)THEN                                00600
        SQR=MSQ-577                                                         00610
      ELSE                                                                  00620
        GOTO 900                                                            00630
      ENDIF                                                                 00640
      B10MSQ=(9-SQR/36)*36 +(71-MOD(IABS(SQR),36))                          00650
     +                     -(71-MOD(IABS(SQR),36))/39*36 -2                 00660
```

```
      RETURN                                                                    00670
  900 B10MSQ=-1                                                                 00680
      RETURN                                                                    00690
      END                                                                       00700
C     ===1=========2=========3=========4=========5=========6=========7==       00710
      LOGICAL FUNCTION B1026(B2,B26,B10)                                        00720
C     -----------FALSE IF 1>@B10>648, ELSE TRUE SUCH THAT @B2 CONTAINS         00730
C                THE 25 BOX2 CONTAINED BY BOX10 @B10 IN NUMERICAL ORDER,      00740
C                AND @B26 CONTAINS ZERO OR THE 26TH BOX2 FOR THE POLAR        00750
C                BOX10.                                                       00760
      IMPLICIT INTEGER(A-Z)                                                     00770
      LOGICAL XYB10,B2XY0                                                       00780
      DIMENSION B2(25)                                                          00790
      JB=B26=0                                                                  00800
      B1026=.FALSE.                                                             00810
      IF(.NOT.XYB10(X1,Y2,B10)) RETURN                                          00820
      X2=X1+80                                                                  00830
      Y1=Y2+80                                                                  00840
      DO 500 Y=Y1,Y2,-20                                                        00850
      DO 500 X=X1,X2, 20                                                        00860
        IF(.NOT.B2XY0(X,Y,BOX2)) RETURN                                         00870
        JB=JB+1                                                                 00880
        B2(JB)=BOX2                                                             00890
  500 CONTINUE                                                                  00900
      IF(B10.EQ.  1) B26=    1                                                  00910
      IF(B10.EQ.648) B26=16202                                                  00920
      B1026=.TRUE.                                                              00930
      RETURN                                                                    00940
      END                                                                       00950
C     ===1=========2=========3=========4=========5=========6=========7==       00960
      LOGICAL FUNCTION B10XY0(X,Y,B10)                                          00970
C     -----------PERFORM <BQXY0> ON 10 DEGREE BOX CORNER @X,@Y                  00980
      IMPLICIT INTEGER(A-E,G-Z)                                                 00990
      LOGICAL BQXY0                                                             01000
      DATA Q/100/,XDIM/36/,Y1/800/,YMOVE/8/,X2/3500/                            01010
C      -- SHIFT LATITUDE X 30 DEGREES WEST TO COMPUTE USING BQXY0               01020
      IF (X .GE. 300) THEN                                                      01030
        XS=X-300                                                                01040
      ELSE                                                                      01050
        XS=X+3300                                                               01060
      ENDIF                                                                     01070
      B10XY0=BQXY0(XS,Y,B10,Q,XDIM,Y1,YMOVE,X2)                                 01080
C      -- SUBTRACT 1 FROM BOX # TO ADJUST FOR LACK OF NORTH POLAR BOX           01090
      B10=B10-1                                                                 01100
      RETURN                                                                    01110
      END                                                                       01120
C     ===1=========2=========3=========4=========5=========6=========7==       01130
C     *F45V1P0*                                                                 01140
      LOGICAL FUNCTION B2XY0(X,Y,B2)                                            01150
C     -----------PERFORM <BQXY0> ON 2 DEGREE BOX CORNER @X,@Y                   01160
      IMPLICIT INTEGER(A-E,G-Z)                                                 01170
      LOGICAL BQXY0                                                             01180
      DATA Q/20/,XDIM/180/,Y1/880/,YMOVE/44/,X2/3580/                           01190
      B2XY0=BQXY0(X,Y,B2,Q,XDIM,Y1,YMOVE,X2)                                    01200
      RETURN                                                                    01210
      END                                                                       01220
```

```
C     ===1=========2=========3=========4=========5=========6=========7==   01230
C     *F45V1P0*                                                           01240
      LOGICAL FUNCTION B4XY0(X,Y,B4)                                      01250
C      ---- FALSE IF @X,@Y ARE NOT THE LOWER-LEFT (SW) CORNER OF A        01260
C           @@/10 DEGREE BOX IN 10THS DEGREE +N,-S,E.                     01270
C           ELSE TRUE RETURNING THE BOX NUMBER @B4                        01280
C           WHERE @XDIM  IS THE NUMBER OF BOXES PER LAT ZONE              01290
C                 @Y1    IS 900-@@                                        01300
C                 @X2    IS THE LARGEST X                                 01310
C                                                                         01320
C       WARNING - DO NOT USE THIS FUNCTION FOR THE POLAR BOXES.           01330
C                 <B4XY0> CANNOT RECOGNIZE (0,900) AS THE SOUTHWEST       01340
C                 CORNER OF THE NORTH POLAR BOX, AND ALL BOXES IN THE     01350
C                 -85 TO -90 DEGREE LATITUDE BAND HAVE (@X,@Y)=(0,-900)   01360
C                 AS THEIR SOUTHWEST CORNER.  THUS <B4XY0> CANNOT TELL    01370
C                 WHICH BOX IS THE SOUTH POLAR BOX WHEN GIVEN (0,-900).   01380
C                                                                         01390
C                 <B4XY0> RETURNS .FALSE. FOR NORTH POLAR BOX.            01400
C                         RETURNS .TRUE. FOR SOUTH POLAR BOX; BUT         01410
C                         THE RETURNED BOX IS NOT THE SOUTH POLAR         01420
C                         BOX.                                            01430
C                                                                         01440
      IMPLICIT INTEGER(A-E,G-Z)                                           01450
      DATA Q/40/,XDIM/90/,Y1/860/,X2/3560/                                01460
      IF(MOD(X,Q).EQ.0.AND.MOD(900-Y,Q).EQ.0.AND.                         01470
     +  (X.GE.0.AND.X.LE.X2)                    .AND.                    01480
     +  (Y.GE.-900.AND.Y.LE.Y1)) GOTO 200                                 01490
      B4XY0=.FALSE.                                                       01500
      RETURN                                                              01510
  200 B4=((900-Y)/Q-1)*XDIM+X/Q+2                                         01520
      B4XY0=.TRUE.                                                        01530
      RETURN                                                              01540
      END                                                                 01550
C     ===1=========2=========3=========4=========5=========6=========7==   01560
C     *F45V1P0*                                                           01570
      LOGICAL FUNCTION BQXY0(X,Y,BQ,Q,XDIM,Y1,YMOVE,X2)                   01580
C     ------FALSE IF @X,@Y ARE NOT THE LOWER-LEFT (SW) CORNER OF A @@/10   01590
C           DEGREE BOX IN 10THS DEGREE +N,-S,E; EXCLUDING POLAR BOXES     01600
C           ELSE TRUE RETURNING THE BOX NUMBER @BQ                        01610
C           WHERE @XDIM  IS THE NUMBER OF BOXES PER LAT ZONE              01620
C                 @Y1    IS 900-@@                                        01630
C                 @YMOVE IS (900/@@)-1                                    01640
C                 @X2    IS THE LARGEST X                                 01650
C                                                                         01660
C       WARNING - DO NOT USE THIS FUNCTION FOR THE POLAR BOXES.           01670
C                 <BQXY0> CANNOT RECOGNIZE (0,900) AS THE SOUTHWEST       01680
C                 CORNER OF THE NORTH POLAR BOX, AND ALL BOXES IN THE     01690
C                 -85 TO -90 DEGREE LATITUDE BAND HAVE (@X,@Y)=(0,-900)   01700
C                 AS THEIR SOUTHWEST CORNER.  THUS <BQXY0> CANNOT TELL    01710
C                 WHICH BOX IS THE SOUTH POLAR BOX WHEN GIVEN (0,-900).   01720
C                                                                         01730
C                 <BQXY0> RETURNS .FALSE. FOR NORTH POLAR BOX.            01740
C                         RETURNS .TRUE. FOR SOUTH POLAR BOX; BUT         01750
C                         THE RETURNED BOX IS NOT THE SOUTH POLAR         01760
C                         BOX.                                            01770
C                                                                         01780
```

```
      IMPLICIT INTEGER(A-E,G-Z)                                          01790
      IF(MOD(X,Q).EQ.0.AND.MOD(Y,Q).EQ.0.AND.                            01800
     +  (X.GE.0.AND.X.LE.X2)              .AND.                         01810
     +  (Y.GE.-900.AND.Y.LE.Y1)) GOTO 200                                01820
      BQXYO=.FALSE.                                                      01830
      RETURN                                                             01840
  200 BQ=(YMOVE-Y/Q)*XDIM+X/Q+2                                          01850
      BQXYO=.TRUE.                                                       01860
      RETURN                                                             01870
C     **  THIS PROGRAM VALID ON FTN4 AND FTN5 **                         01880
      END                                                                01890
C     ===1=========2=========3=========4=========5=========6=========7==  01900
      INTEGER FUNCTION MSQB10(B10)                                       01910
C     -----------EQUALS -1 IF ILLEGAL B10, ELSE EQUALS EQUIVALENT MSQ    01920
      IMPLICIT INTEGER(A-E,G-Z)                                          01930
      MSQB10=-1                                                          01940
      M=MOD(B10,36)                                                      01950
      IF (M .EQ. 0) M=36                                                 01960
      IF (B10 .GE. 1 .AND. B10 .LE. 33) THEN                             01970
         MSQB10 = 934-B10                                                01980
       ELSE                                                              01990
         MSQB10 = 970-B10                                                02000
      ENDIF                                                              02010
      IF (B10 .GE. 37 .AND. B10 .LE. 324) THEN                           02020
       IF (M .GE. 1 .AND. M .LE. 33) THEN                                02030
          MSQB10 = 322-B10                                               02040
        ELSE                                                             02050
          MSQB10 = 358-B10                                               02060
      ENDIF                                                              02070
      ENDIF                                                              02080
      IF (B10 .GE. 325 .AND. B10 .LE. 648) THEN                          02090
       IF (M .GE. 1 .AND. M .LE. 33) THEN                                02100
          MSQB10 = 333-M+((AINT(B10/36.0)-9)*36)                         02110
        ELSE IF (M .EQ. 34 .OR. M .EQ. 35) THEN                          02120
            MSQB10 = 369-M+((AINT(B10/36.0)-9)*36)                       02130
          ELSE IF (M .EQ. 36) THEN                                       02140
             MSQB10 = 333+((AINT(B10/36.0)-10)*36)                       02150
        ENDIF                                                            02160
      ENDIF                                                              02170
      RETURN                                                             02180
      END                                                                02190
C     ===1=========2=========3=========4=========5=========6=========7==  02200
C     *F45V1P0*                                                          02210
      LOGICAL FUNCTION MSQXYO(X,Y,MSQ)                                   02220
C     -----------RETURNS MSQ BOX# @MSQ GIVEN 10 DEGREE BOX CORNER @X, @Y  02230
C                RETURNS FALSE IF @X,@Y IS NOT THE CORNER OF A 10 DEGREE  02240
C                BOX.                                                    02250
C                                                                        02260
C                <MSQXYO> USES <BQXYO> - SEE WARNING BELOW.              02270
C                                                                        02280
C       WARNING - DO NOT USE THIS FUNCTION FOR THE POLAR BOXES.          02290
C                 <BQXYO> CANNOT RECOGNIZE (0,900) AS THE SOUTHWEST      02300
C                 CORNER OF THE NORTH POLAR BOX, AND ALL BOXES IN THE    02310
C                 -85 TO -90 DEGREE LATITUDE BAND HAVE (@X,@Y)=(0,-900)  02320
C                 AS THEIR SOUTHWEST CORNER.  THUS <BQXYO> CANNOT TELL   02330
C                 WHICH BOX IS THE SOUTH POLAR BOX WHEN GIVEN (0,-900).  02340
```

```
C                                                                       02350
C                 <BQXYO> RETURNS .FALSE. FOR NORTH POLAR BOX.          02360
C                         RETURNS .TRUE. FOR SOUTH POLAR BOX; BUT       02370
C                         THE RETURNED BOX IS NOT THE SOUTH POLAR       02380
C                         BOX.                                          02390
C                                                                       02400
      IMPLICIT INTEGER(A-E,G-Z)                                         02410
      LOGICAL BQXYO                                                     02420
C      -- SHIFT LATITUDE X 30 DEGREES WEST TO COMPUTE USING BQXYO       02430
      IF (X .GE. 300) THEN                                              02440
        XS=X-300                                                        02450
      ELSE                                                              02460
        XS=X+3300                                                       02470
      ENDIF                                                             02480
      DATA Q/100/,XDIM/36/,Y1/800/,YMOVE/8/,X2/3500/                    02490
      MSQXYO=BQXYO(XS,Y,BQ,Q,XDIM,Y1,YMOVE,X2)                          02500
C      -- SUBTRACT 1 FROM BOX # TO ADJUST FOR LACK OF POLAR BOX AND     02510
C         RECALCULATE THE EQUIVALENT MARSDEN SQUARE                     02520
      MSQ=MSQB10(BQ-1)                                                  02530
      RETURN                                                            02540
      END                                                               02550
C     ===1=========2=========3=========4=========5=========6=========7==  02560
      INTEGER FUNCTION QCDCXY(X,Y)                                      02570
C     ------RETURNS -1 UNLESS 900<@Y<-900, 3599<@X<0, @X<>1800 (10THS E)  02580
C           RETURNS THE NCDC QUADRANT 1=NW,2=NE,3=SW,4=SE OTHERWISE     02590
      IMPLICIT INTEGER(A-E,G-Z)                                         02600
      IF(Y.LT.900.AND.Y.GT.-900.AND.X.LT.3599.AND.X.GT.0.AND.X.NE.1800)  02610
     + THEN                                                             02620
        QCDCXY=1                                                        02630
        IF(X.LT.1800) QCDCXY=QCDCXY+1                                   02640
        IF(Y.LT.0)    QCDCXY=QCDCXY+2                                   02650
      ELSE                                                              02660
        QCDCXY=-1                                                       02670
      ENDIF                                                             02680
      RETURN                                                            02690
      END                                                               02700
C     ===1=========2=========3=========4=========5=========6=========7==  02710
      LOGICAL FUNCTION XYB10(X,Y,B10)                                   02720
C     -----------PERFORM <XYBQ> ON A 10 DEGREE BOX @B10                02730
      IMPLICIT INTEGER(A-E,G-Z)                                         02740
      LOGICAL XYBQ                                                      02750
      DATA Q/100/,LAST/648/,XDIM/36/,Y1/800/,POLE/1/,XMOVE/300/         02760
      XYB10=XYBQ(X,Y,B10,Q,LAST,XDIM,Y1,POLE,XMOVE)                     02770
      RETURN                                                            02780
      END                                                               02790
C     ===1=========2=========3=========4=========5=========6=========7==  02800
C     *F45V1P0*                                                         02810
      LOGICAL FUNCTION XYB2(X,Y,B2)                                     02820
C     -----------PERFORM <XYBQ> ON A 2 DEGREE BOX @B2                  02830
      IMPLICIT INTEGER(A-E,G-Z)                                         02840
      LOGICAL XYBQ                                                      02850
      DATA Q/20/,LAST/16202/,XDIM/180/,Y1/880/,POLE/2/,XMOVE/0/         02860
      XYB2=XYBQ(X,Y,B2,Q,LAST,XDIM,Y1,POLE,XMOVE)                       02870
      RETURN                                                            02880
      END                                                               02890
C     ===1=========2=========3=========4=========5=========6=========7==  02900
```

```
C     *F45V1P0*                                                          02910
      LOGICAL FUNCTION XYB4(X,Y,B4)                                      02920
C     -----------PERFORM <XYBQ> ON A 4 DEGREE BOX @B4                    02930
      IMPLICIT INTEGER(A-E,G-Z)                                          02940
      LOGICAL XYBQ                                                       02950
      DATA Q/40/,LAST/4052/,XDIM/90/,Y1/860/,POLE/2/,XMOVE/0/            02960
      XYB4=XYBQ(X,Y,B4,Q,LAST,XDIM,Y1,POLE,XMOVE)                        02970
      RETURN                                                             02980
      END                                                                02990
C     ===1=========2=========3=========4=========5=========6=========7==  03000
C     *F45V1P0*                                                          03010
      LOGICAL FUNCTION XYBQ(X,Y,BQ,Q,LAST,XDIM,Y1,POLE,XMOVE)            03020
C     ------FALSE IF 1>BQ>@LAST, ELSE TRUE SUCH THAT @X,@Y ARE THE       03030
C           LAT,LON IN 10THS DEGREE +N,-S,E OF LOWER-LEFT (SW) CORNER    03040
C           OF @Q/10 DEGREE BOX @BQ;  POLAR @X ARE SET TO 0              03050
C           WHERE @LAST  IS THE LAST BOX NUMBER                          03060
C                 @XDIM  IS THE NUMBER OF BOXES PER LAT ZONE             03070
C                 @Y1    IS 900-@Q                                       03080
C                 @POLE  IS 1 IF 0 POLAR BOXES, 2 IF 2 POLAR BOXES       03090
C                 @XMOVE IS THE X-ORIGIN                                 03100
      IMPLICIT INTEGER(A-E,G-Z)                                          03110
      XYBQ=.FALSE.                                                       03120
      IF(BQ.LT.1.OR.BQ.GT.LAST) RETURN                                   03130
      IF(POLE.EQ.1) GOTO 200                                             03140
      IF(BQ.NE.1) GOTO 100                                               03150
      X=0                                                                03160
      Y= 900                                                             03170
      GOTO 900                                                           03180
  100 IF(BQ.NE.LAST) GOTO 200                                            03190
      X=0                                                                03200
      Y=-900                                                             03210
      GOTO 900                                                           03220
  200 CONTINUE                                                           03230
      X=MOD(BQ-POLE,XDIM)*Q+XMOVE                                        03240
      IF(X.GE.3600) X=X-3600                                             03250
      Y=Y1-(BQ-POLE)/XDIM*Q                                              03260
  900 XYBQ=.TRUE.                                                        03270
      RETURN                                                             03280
C     **  THIS PROGRAM VALID ON FTN4 AND FTN5 **                         03290
      END                                                                03300
C     ===1=========2=========3=========4=========5=========6=========7==  03310
      LOGICAL FUNCTION XYMSQ(X,Y,MSQ)                                    03320
C     ---------- PERFORM <B10MSQ> TO CONVERT @MSQ TO @B10, THEN USES     03330
C                <XYBQ> TO FIND LAT. AND LONG. OF EQUIVALENT @B10        03340
      IMPLICIT INTEGER(A-E,G-Z)                                          03350
      LOGICAL XYBQ                                                       03360
      B10 = B10MSQ(MSQ)                                                  03370
      DATA Q/100/,LAST/648/,XDIM/36/,Y1/800/,POLE/1/,XMOVE/300/          03380
      XYMSQ=XYBQ(X,Y,B10,Q,LAST,XDIM,Y1,POLE,XMOVE)                      03390
      RETURN                                                             03400
      END                                                                03410
```

```
      PROGRAM QI9
C-----READ AND PRINT MSU2
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01G. SL     85/01/24.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=100,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=1600,ID=0
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=117)
C
      COMMON /MSU2/FUNITS(117),FBASE(117),BITS(117),OFFSET(117)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(8,14)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01G/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QI9',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      PRINT 300,FTRUE
300   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0,' BOX10 ',F4.0
     +,' CHECKSUM ',F6.0/
     +8X,'S',7X,'A',7X,'W',7X,'U',7X,'V',7X,'P',7X,'C',7X,'Q'/
     +1X,'D',8F8.1/
     +1X,'H',8F8.1/
     +1X,'X',8F8.2/
     +1X,'Y',8F8.2/
     +1X,'N',8F8.0/
     +1X,'M',6F8.2,F8.1,F8.2/
     +1X,'S',6F8.2,F8.1,F8.2/
     +1X,'0',6F8.2,F8.1,F8.2/
     +1X,'1',6F8.2,F8.1,F8.2/
     +1X,'2',6F8.2,F8.1,F8.2/
     +1X,'3',6F8.2,F8.1,F8.2/
```

```
     +1X,'4',6F8.2,F8.1,F8.2/
     +1X,'5',6F8.2,F8.1,F8.2/
     +1X,'6',6F8.2,F8.1,F8.2)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      BLOCK DATA MSU2
      IMPLICIT INTEGER(A-E,G-Z)
      COMMON /MSU2/FUNITS(117),FBASE(117),BITS(117),OFFSET(117)
C
      DATA FUNITS/5*1.
     +,8*.2,8*.1,16*.01,8*1.
     +,6*.01,.1,.01
     +,6*.01,.1,.01
     +,6*.01,.1,.01
     +,6*.01,.1,.01
     +,6*.01,.1,.01
     +,6*.01,.1,.01
     +,6*.01,.1,.01
     +,6*.01,.1,.01/
C
      DATA FBASE/1799,4*0
     +,8*4,24*-1,8*0,-501,-8801,-1,2*-10221,86999,2*-1,8*-1
     +,-501,-8801,-1,2*-10221,86999,2*-1
     +,-501,-8801,-1,2*-10221,86999,2*-1
     +,-501,-8801,-1,2*-10221,86999,2*-1
     +,-501,-8801,-1,2*-10221,86999,2*-1
     +,-501,-8801,-1,2*-10221,86999,2*-1
     +,-501,-8801,-1,2*-10221,86999,2*-1/
C
      DATA BITS/8,4,14,10,12,32*8,80*16/
C
      DATA OFFSET/
     +   16,  24,  28,  42,  52,  64,  72,  80,  88,  96, 104, 112, 120
     +, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224
     +, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 336
     +, 352, 368, 384, 400, 416, 432, 448, 464, 480, 496, 512, 528, 544
     +, 560, 576, 592, 608, 624, 640, 656, 672, 688, 704, 720, 736, 752
     +, 768, 784, 800, 816, 832, 848, 864, 880, 896, 912, 928, 944, 960
     +, 976, 992,1008,1024,1040,1056,1072,1088,1104,1120,1136,1152,1168
     +,1184,1200,1216,1232,1248,1264,1280,1296,1312,1328,1344,1360,1376
     +,1392,1408,1424,1440,1456,1472,1488,1504,1520,1536,1552,1568,1584/
      END
C=====================================================================
      SUBROUTINE GETRPT(TAPE,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C
C-----RETURN FLOATING POINT VALUES IN FTRUE
C
C     INPUT
C         TAPE - RPTIN/RCDIN UNIT
C         FMISS - MISSING VALUE
```

```
C          FUNITS(DIM UN) - UNITS FOR UNCODING
C          FBASE(DIM UN) - BASE FOR UNCODING
C          BITS(DIM UN) - BITS FOR UNPACKING
C          OFFSET(DIM UN) - OFFSET FOR UNPACKING
C          INDEXCK - UN(INDEXCK) = CHECKSUM
C          ID - GROUP NUMBER FOR IDENTIFICATION CHECKSUM
C          BPR - BITS PER REPORT
C          BPW - BITS PER WORD
C          RPTOFF - 0=FALSE 1=TRUE
C      OUTPUT
C          BUF(DIM BUF) - RPTIN/RCDIN BUFFER
C          PK(DIM PK) - PACKED REPORT
C          UN(DIM UN) - UNPACKED REPORT
C          FTRUE(DIM UN) - TRUE VALUES
C          JEOF - 0=FALSE 1=TRUE
C
       IMPLICIT INTEGER(A-E,G-Z)
       DIMENSION FUNITS(DIM UN),FBASE(DIM UN),BITS(DIM UN),OFFSET(DIM UN)
      +,BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----RPTIN/RCDIN
       IF(RPTOFF.NE.0)GOTO 100
       CALL RPTIN(TAPE,BUF,PK,KWDS,1,DIM PK,JEOF)
       GOTO 110
100    CALL RCDIN(TAPE,BUF,DIM BUF,PK,DIM PK,BPR,BPW,JEOF)
110    IF(JEOF-1)200,900,800
C
C-----GBYTE AND CONVERT TO TRUE
200    CK=ID
       DO 230 I=1,DIM UN
       CALL GBYTE(PK(OFFSET(I)/BPW+1),UN(I),MOD(OFFSET(I),BPW),BITS(I))
       IF(I.EQ.INDEXCK)GOTO 210
       IF(UN(I).EQ.0)GOTO 220
       FTRUE(I)=(UN(I)+FBASE(I))*FUNITS(I)
       CK=CK+UN(I)
       GOTO 230
210    FTRUE(INDEXCK)=UN(INDEXCK)
       GOTO 230
220    FTRUE(I)=FMISS
230    CONTINUE
       IF(MOD(CK,2**BITS(INDEXCK)-1).EQ.UN(INDEXCK))RETURN
C
C-----ERROR
       PRINT *,' SUBROUTINE GETRPT  --  CHECKSUM ERROR, TAPE = ',TAPE
      +,', REPORT = ',BUF(2)
       PRINT *,' FTRUE = ',FTRUE
800    STOP
C
900    END
C=====================================================================
       SUBROUTINE RCDIN(TAPE,BUF,DIM BUF,RCD,DIM RCD,BPR,BPW,JEOF)
C
C-----RETURN ONE LOGICAL RECORD IN RCD
C
C      INPUT
C          TAPE - BUFFER IN UNIT
```

```
C          BPR - BITS PER RECORD
C          BPW - BITS PER WORD
C      OUTPUT
C          BUF(DIM BUF) - PHYSICAL RECORD
C          RCD(DIM RCD) - LOGICAL RECORD
C          JEOF - 0=FALSE 1=TRUE
C
C      BUF(1) = GBYTE OFFSET
C      BUF(2) = LOGICAL RECORD COUNT
C      BUF(3) = PHYSICAL RECORD COUNT
C      BUF(4) =
C      BUF(5) = BLOCK LENGTH IN BITS
C      BUF(6) =
C
       IMPLICIT INTEGER(A-E,G-Z)
       REAL UNIT
       DIMENSION BUF(DIM BUF),RCD(DIM RCD)
C
       IF(BUF(1)+BPR.LE.BUF(5))GOTO 200
C-----BUFFER IN
10     BUFFER IN(TAPE,1)(BUF(7),BUF(DIM BUF))
       JEOF=UNIT(TAPE)+1
       IF(JEOF-1)100,100,800
100    BUF(1)=0
       BUF(5)=LENGTH(TAPE)*BPW
       IF(JEOF.EQ.1)RETURN
       BUF(3)=BUF(3)+1
C
C-----GBYTE
200    CALL GBYTES
      +(BUF(6+BUF(1)/BPW+1) ,RCD ,MOD(BUF(1),BPW) ,BPW ,0 ,DIM RCD)
       IF(RCD(1).EQ.0.AND.RCD(2).EQ.0)GOTO 10
       BUF(1)=BUF(1)+BPR
       BUF(2)=BUF(2)+1
       RETURN
C
C-----ERROR
800    PRINT *,' SUBROUTINE RCDIN  --  BUFFER IN ERROR, TAPE = ',TAPE
      +,', BLOCK = ',BUF(3)+1
       STOP
       END
```

```
      PROGRAM QI12
C-----READ AND PRINT CMR4
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01D. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=300,RPTOFF=1,FMISS=-999.9,INDEXCK=30,BPR=192,ID=0
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=30)
C
      COMMON /CMR4/FIELD(30),FTRUEL(30),FTRUEU(30),FUNITS(30)
     +,FBASE(30),BITS(30),OFFSET(30)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
      DATA LEVEL/4H.01D/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QI12',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      PRINT 300,(FIELD(I),FTRUE(I),I=1,DIM UN)
300   FORMAT(6(1X,A5,F7.1))
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      BLOCK DATA CMR4
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /CMR4/FIELD(30),FTRUEL(30),FTRUEU(30),FUNITS(30)
     +,FBASE(30),BITS(30),OFFSET(30)
C
      DATA FIELD/
     +8HBOX10    ,8HMONTH    ,8HBOX2     ,8HYEAR     ,8HDAY      ,
     +8HHOUR     ,8HX        ,8HY        ,8HS        ,8HBI       ,
     +8HA        ,8HDP       ,8HTI       ,8HW        ,8HWI       ,
     +8HU        ,8HV        ,8HDI       ,8HP        ,8HC        ,
```

```
     +8HNH       ,8HCL       ,8HH        ,8HHI       ,8HCM       ,
     +8HCH       ,8HST       ,8HPW       ,8HCD       ,8HCK       /
C
      DATA FTRUEL/
     +3*1.,1800.,1.,3*0.,-5.,0.,-88.,4*0.,2*-102.2,0.,870.,11*0./
C
      DATA FTRUEU/
     +648.,12.,16202.,2054.,31.,23.,2*2.,40.,2.,58.,70.,5.,102.2,1.
     +,2*102.2,5.,1074.6,2*9.,2*10.,1.,2*10.,7.,99.,999.,62./
C
      DATA FUNITS/
     +6*1.,3*.1,1.,2*.1,1.,.1,1.,2*.1,1.,.1,11*1./
C
      DATA FBASE/
     +3*0,1799,0,3*-1,-51,-1,-881,4*-1,2*-1023,-1,8699,10*-1,0/
C
      DATA BITS/
     +10,4,14,8,4*5,9,2,11,10,3,10,2,2*11,3,11,4*4,2,3*4,7,10,6/
C
      DATA OFFSET/
     +   0, 10, 14, 28, 36, 41, 46, 51, 56, 65, 67, 78, 88, 91,101
     +,103,114,125,128,139,143,147,151,155,157,161,165,169,176,186/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QI21
C-----READ AND PRINT MSUG1 GROUP1
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01D. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=1
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSUG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01D/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QI21',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSUG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=======================================================================
      SUBROUTINE WRMSUG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +8X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'S',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
```

```
     +1X,'A',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
     +1X,'P',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
     +1X,'Q',2F8.2,F8.0,F8.2,2F8.0,2F8.1)
      END
C=====================GROUP 1=========================================
      BLOCK DATA MSUG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSUG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/5*1.
     +,4*.01
     +,4*.01
     +,4*1.
     +,4*.01
     +,4*2.
     +,4*2.
     +,4*.2
     +,4*.2/
C
      DATA FBASE/1799,4*0
     +,-501.,-8801.,86999.,-1.
     +,-501.,-8801.,86999.,-1.
     +,4*0.
     +,4*-1.
     +,4*0.
     +,4*-.5
     +,4*-.5
     +,4*-.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16, 24, 28, 42, 52, 64, 80, 96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=======================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QI22
C-----READ AND PRINT MSUG1 GROUP 2
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01D. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=2
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSUG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01D/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QI22',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSUG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      SUBROUTINE WRMSUG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +8X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'W',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
```

```
     +1X,'U',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
     +1X,'V',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
     +1X,'C',2F8.1,F8.0,F8.1,2F8.0,2F8.1)
      END
C=====================GROUP 2=========================================
      BLOCK DATA MSUG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSUG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/5*1.
     +,3*.01,.1
     +,3*.01,.1
     +,4*1.
     +,3*.01,.1
     +,4*2.
     +,4*2.
     +,4*.2
     +,4*.2/
C
      DATA FBASE/1799,4*0
     +,-1.,2*-10221.,-1.
     +,-1.,2*-10221.,-1.
     +,4*0.
     +,4*-1.
     +,4*0.
     +,4*-.5
     +,4*-.5
     +,4*-.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16, 24, 28, 42, 52, 64, 80, 96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=======================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QI24
C-----READ AND PRINT DSU2
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=250,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=960,ID=0
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=58)
C
      COMMON /DSU2/FUNITS(58),FBASE(58),BITS(58),OFFSET(58)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(8,6)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QI24',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      PRINT 300,FTRUE
300   FORMAT(/' DECADE ',F4.0,' MONTH ',F3.0,' BOX2 ',F6.0,' BOX10 '
     +,F4.0,' CHECKSUM ',F6.0/
     +8X,'0',7X,'1',7X,'2',7X,'3',7X,'4',7X,'5',7X,'6',7X,'N'/
     +1X,'S',7F8.2,F8.0/
     +1X,'A',7F8.2,F8.0/
     +1X,'U',7F8.2,F8.0/
     +1X,'V',7F8.2,F8.0/
     +1X,'P',7F8.2,F8.0/
     +1X,'R',7F8.1,F8.0/
     +1X,' U ',F8.2,' V ',F8.2,' UV ',F8.2,' UU ',F8.2,' VV ',F8.2)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
```

```
C=======================================================================
      BLOCK DATA DSU2
      IMPLICIT INTEGER(A-E,G-Z)
      COMMON /DSU2/FUNITS(58),FBASE(58),BITS(58),OFFSET(58)
C
      DATA FUNITS/5*1.
     +,7*.01,1. ,7*.01,1. ,7*.01,1. ,7*.01,1. ,7*.01,1. ,7*.1,1.
     +,5*.01/
C
     +,FBASE/179,4*0
     +,7*-501,0 ,7*-8801,0 ,7*-10221,0 ,7*-10221,0 ,7*86999,0 ,7*-1,0
     +,2*-10221,-522243,2*-1/
C
     +,BITS/8,4,14,10,12,50*16,3*32/
C
     +,OFFSET/
     +  16, 24, 28, 42, 52, 64, 80, 96,112,128,144,160,176,192,208,224
     +,240,256,272,288,304,320,336,352,368,384,400,416,432,448,464,480
     +,496,512,528,544,560,576,592,608,624,640,656,672,688,704,720,736
     +,752,768,784,800,816,832,848,864,896,928/
      END
C=======================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL14
C-----READ AND PRINT MST3
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=60,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=3712,ID=0
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=271)
C
      COMMON /MST3/FUNITS(271),FBASE(271),BITS(271),OFFSET(271)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(19,14)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL14',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      PRINT 300,(FTRUE(I),I=1,5)
300   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +9X,7X,'D',7X,'H',7X,'X',7X,'Y',7X,'N',7X,'M',7X,'S'
     +  ,7X,'0',7X,'1',7X,'2',7X,'3',7X,'4',7X,'5',7X,'6')
      PRINT 301,((FTRUE2(I,J),J=1,14),I=1,19)
301   FORMAT(1X,'S        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'A        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'W        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'U        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'V        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'P        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'C        ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'Q        ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'R        ',F8.1,3F8.2,F8.0,9F8.1/
```

```
     +1X,'S-A      ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'(S-A)*W  ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'QS-Q     ',F8.1,3F8.2,F8.0,9F8.2/
     +1X,'(QS-Q)*W',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'W*U      ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'W*V      ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'U*A      ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'V*A      ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'U*Q      ',F8.1,3F8.2,F8.0,9F8.1/
     +1X,'V*Q      ',F8.1,3F8.2,F8.0,9F8.1)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C======================================================================
      BLOCK DATA MST3
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MST3/FUNITS(271),FBASE(271),BITS(271),OFFSET(271)
C
      DATA FUNITS/5*1.
     +,19*.2,57*.01,19*1.
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1
     +,6*.01,.1,.01,.1,.01,.1,.01,7*.1/
C
      DATA FBASE/1799,4*0
     +,19*4,57*-1,19*0
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,19*-1
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001
     +,-501,-8801,-1,2*-10221,86999,3*-1
     +,-6301,-10001,-4001,-10001,2*-30001,2*-20001,2*-10001/
C
      DATA BITS/8,4,14,10,12,76*8,190*16/
C
      DATA OFFSET/ 16, 24, 28, 42, 52, 64
     +, 72, 80, 88, 96,104,112,120,128,136,144,152,160,168,176,184,192
```

```
     +,200,208,216,224,232,240,248,256,264,272,280,288,296,304,312,320
     +,328,336,344,352,360,368,376,384,392,400,408,416,424,432,440,448
     +,456,464,472,480,488,496,504,512,520,528,536,544,552,560,568,576
     +,584,592,600,608,616,624,632,640,648,656,664,672,688,704,720,736
     +,752,768,784,800,816,832,848,864,880,896,912,928,944,960,976,992
     +,1008,1024,1040,1056,1072,1088,1104,1120,1136,1152,1168,1184,1200
     +,1216,1232,1248,1264,1280,1296,1312,1328,1344,1360,1376,1392,1408
     +,1424,1440,1456,1472,1488,1504,1520,1536,1552,1568,1584,1600,1616
     +,1632,1648,1664,1680,1696,1712,1728,1744,1760,1776,1792,1808,1824
     +,1840,1856,1872,1888,1904,1920,1936,1952,1968,1984,2000,2016,2032
     +,2048,2064,2080,2096,2112,2128,2144,2160,2176,2192,2208,2224,2240
     +,2256,2272,2288,2304,2320,2336,2352,2368,2384,2400,2416,2432,2448
     +,2464,2480,2496,2512,2528,2544,2560,2576,2592,2608,2624,2640,2656
     +,2672,2688,2704,2720,2736,2752,2768,2784,2800,2816,2832,2848,2864
     +,2880,2896,2912,2928,2944,2960,2976,2992,3008,3024,3040,3056,3072
     +,3088,3104,3120,3136,3152,3168,3184,3200,3216,3232,3248,3264,3280
     +,3296,3312,3328,3344,3360,3376,3392,3408,3424,3440,3456,3472,3488
     +,3504,3520,3536,3552,3568,3584,3600,3616,3632,3648,3664,3680,3696/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL16
C-----READ AND PRINT TRP1
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=250,RPTOFF=1,FMISS=0.,INDEXCK=5,BPR=256,ID=0
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=23)
C
      COMMON /TRP1/FUNITS(23),FBASE(23),BITS(23),OFFSET(23)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(6,3)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL16',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      PRINT 300,FTRUE
300   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0,' BOX10 ',F4.0
     +,' CHECKSUM ',F5.0/
     +,9X,'S ',6X,'A ',6X,'U ',6X,'V ',6X,'P ',6X,'R '/
     +,1X,'NI',6F8.0/
     +,1X,'NL',6F8.0/
     +,1X,'NU',6F8.0)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      BLOCK DATA TRP1
      IMPLICIT INTEGER(A-E,G-Z)
C
```

```
      COMMON /TRP1/FUNITS(23),FBASE(23),BITS(23),OFFSET(23)
C
      DATA FUNITS/5*1.
     +,18*1./
C
     +,FBASE/1799,4*0
     +,18*0/
C
     +,BITS/8,4,14,10,12
     +,6*12,12*10/
C
     +,OFFSET/ 16, 24, 28, 42, 52
     +, 64, 76, 88,100,112,124,136,146,156,166,176,186,196,206,216,226
     +,236,246/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL21
C-----READ AND PRINT CMR5
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=300,RPTOFF=1,FMISS=-999.9,INDEXCK=35,BPR=192,ID=0
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=35)
C
      COMMON /CMR5/FIELD(35),FTRUEL(35),FTRUEU(35),FUNITS(35)
     +,FBASE(35),BITS(35),OFFSET(35)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL21',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      PRINT 300,(FIELD(I),FTRUE(I),I=1,DIM UN)
300   FORMAT(6(1X,A5,F7.1))
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      BLOCK DATA CMR5
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /CMR5/FIELD(35),FTRUEL(35),FTRUEU(35),FUNITS(35)
     +,FBASE(35),BITS(35),OFFSET(35)
C
      DATA FIELD/8HBOX10   ,8HMONTH   ,8HBOX2    ,8HYEAR    ,8HDAY     ,
     +8HHOUR    ,8HX       ,8HY       ,8HS       ,8HBI      ,8HA       ,
     +8HDP      ,8HTI      ,8HU       ,8HV       ,8HDI      ,8HWI      ,
     +8HP       ,8HC       ,8HNH      ,8HCL      ,8HH       ,8HHI      ,
     +8HCM      ,8HCH      ,8HST      ,8HPW      ,8HCD      ,8HLF      ,
```

```
     +8HSF       ,8HAF       ,8HRF       ,8HWF       ,8HPF       ,8HCK       /
C
      DATA FTRUEL/3*1.,1800.,1.,3*0.,-5.,0.,-88.,2*0.,2*-102.2,2*0.,870.
     +,17*0./
C
      DATA FTRUEU/648.,12.,16202.,2054.,31.,23.,2*2.,40.,2.,58.,70.,5.
     +,2*102.2,5.,1.,1074.6,2*9.,2*10.,1.,2*10.,7.,99.,999.,0.,5*2.,30./
C
      DATA FUNITS/6*1.,3*.1,1.,2*.1,1.,2*.1,2*1.,.1,17*1./
C
      DATA FBASE/3*0,1799,0,3*-1,-51,-1,-881,2*-1,2*-1023,2*-1,8699
     +,16*-1,0/
C
      DATA BITS/10,4,14,8,4*5,9,2,11,10,3,2*11,3,2,11,4*4,2,3*4,7,10
     +,1,5*2,5/
C
C     RPTOFF 0
C      DATA OFFSET/
C     +  64, 74, 78, 92,100,105,110,115,120,129,131,142,152,155,166,177
C     +,180,182,193,197,201,205,209,211,215,219,223,230,240,241,243,245
C     +,247,249,251/
C
C     RPTOFF 1
      DATA OFFSET/
     +   0, 10, 14, 28, 36, 41, 46, 51, 56, 65, 67, 78, 88, 91,102,113
     +,116,118,129,133,137,141,145,147,151,155,159,166,176,177,179,181
     +,183,185,187/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL28
C-----READ AND PRINT MSTG1 GROUP 3
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     ----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     -----------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=3
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL28',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSTG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      SUBROUTINE WRMSTG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'S        ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
```

```
     +1X,'A       ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
     +1X,'Q       ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
     +1X,'R       ',2F8.1,F8.0,F8.1,F8.0,3F8.1)
      END
C=====================GROUP 3=========================================
      BLOCK DATA MSTG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/1., 1.,  1.,  1.,  1.
     +,1.E-2,  1.E-2,  1.E-2,  0.1
     +,1.E-2,  1.E-2,  1.E-2,  0.1
     +,1.,  1.,  1.,  1.
     +,1.E-2,  1.E-2,  1.E-2,  0.1
     +,2.,  2.,  2.,  2.
     +,0.1,  0.1,  0.1,  0.1
     +,0.2,  0.2,  0.2,  0.2
     +,0.2,  0.2,  0.2,  0.2/
C
      DATA FBASE/1799.,  0.,  0.,  0.,  0.
     +,-501.,  -8801.,  -1.,  -1.
     +,-501.,  -8801.,  -1.,  -1.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,-.5, -.5, -.5, -.5
     +,-.5, -.5, -.5, -.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16, 24, 28, 42, 52, 64, 80, 96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL29
C-----READ AND PRINT MSTG1 GROUP 4
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=4
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL29',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSTG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      SUBROUTINE WRMSTG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'W       ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
```

```
     +1X,'U       ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
     +1X,'V       ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
     +1X,'P       ',2F8.2,F8.0,F8.2,F8.0,3F8.1)
      END
C=====================GROUP 4=======================================
      BLOCK DATA MSTG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/1., 1.,  1.,  1.,  1.
     +,1.E-2,  1.E-2,  1.E-2,  1.E-2
     +,1.E-2,  1.E-2,  1.E-2,  1.E-2
     +,1.,  1.,  1.,  1.
     +,1.E-2,  1.E-2,  1.E-2,  1.E-2
     +,2.,  2.,  2.,  2.
     +,0.1,  0.1,  0.1,  0.1
     +,0.2,  0.2,  0.2,  0.2
     +,0.2,  0.2,  0.2,  0.2/
C
      DATA FBASE/1799.,  0.,  0.,  0.,  0.
     +,-1.,  -10221.,  -10221.,  86999.
     +,-1.,  -10221.,  -10221.,  86999.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,-.5, -.5, -.5, -.5
     +,-.5, -.5, -.5, -.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16, 24, 28, 42, 52, 64, 80, 96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL30
C-----READ AND PRINT MSTG1 GROUP 5
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=5
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL30',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSTG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      SUBROUTINE WRMSTG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'C       ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
```

```
     +1X,'R        ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
     +1X,'W*U      ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
     +1X,'W*V      ',2F8.1,F8.0,F8.1,F8.0,3F8.1)
      END
C=====================GROUP 5=======================================
      BLOCK DATA MSTG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/1., 1.,  1.,  1.,  1.
     +,0.1,  0.1,  0.1,  0.1
     +,0.1,  0.1,  0.1,  0.1
     +,1.,   1.,   1.,   1.
     +,0.1,  0.1,  0.1,  0.1
     +,2.,   2.,   2.,   2.
     +,0.1,  0.1,  0.1,  0.1
     +,0.2,  0.2,  0.2,  0.2
     +,0.2,  0.2,  0.2,  0.2/
C
      DATA FBASE/1799.,  0.,  0.,  0.,  0.
     +,-1.,   -1.,   -30001.,   -30001.
     +,-1.,   -1.,   -30001.,   -30001.
     +,0.,    0.,    0.,    0.
     +,-1.,   -1.,   -1.,   -1.
     +,0.,    0.,    0.,    0.
     +,-1.,   -1.,   -1.,   -1.
     +,-.5,  -.5,  -.5,  -.5
     +,-.5,  -.5,  -.5,  -.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16,  24,  28,  42,  52,  64,  80,  96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL31
C-----READ AND PRINT MSTG1 GROUP 6
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     ----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     -----------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=6
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL31',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSTG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      SUBROUTINE WRMSTG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'S-A      ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
```

```
     +1X,'(S-A)*W ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
     +1X,'QS-Q    ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
     +1X,'(QS-Q)*W',2F8.1,F8.0,F8.1,F8.0,3F8.1)
      END
C=====================GROUP 6=========================================
      BLOCK DATA MSTG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/1., 1.,  1.,  1.,  1.
     +,1.E-2,  0.1,  1.E-2,  0.1
     +,1.E-2,  0.1,  1.E-2,  0.1
     +,1.,  1.,  1.,  1.
     +,1.E-2,  0.1,  1.E-2,  0.1
     +,2.,  2.,  2.,  2.
     +,0.1,  0.1,  0.1,  0.1
     +,0.2,  0.2,  0.2,  0.2
     +,0.2,  0.2,  0.2,  0.2/
C
      DATA FBASE/1799.,  0.,  0.,  0.,  0.
     +,-6301.,  -10001.,  -4001.,  -10001.
     +,-6301.,  -10001.,  -4001.,  -10001.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,-.5, -.5, -.5, -.5
     +,-.5, -.5, -.5, -.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16, 24, 28, 42, 52, 64, 80, 96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
      PROGRAM QL32
C-----READ AND PRINT MSTG1 GROUP 7
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, GBYTE/S, DATE AND TIME ARE
C     MACHINE-DEPENDENT ROUTINES AND FUNCTIONS.  SEE COADS RELEASE 1
C     SUPPLEMENT H FOR A DESCRIPTION OF THEIR BEHAVIOR.  BPW IS A
C     PARAMETER WHICH MUST BE SET TO THE NUMBER OF BITS PER MACHINE
C     WORD.
C     ===1=========2=========3========4=========5=========6=========7==
C
C     -----------REVISION HISTORY---------------------------------------
C      LEVEL AUTHOR DATE       DESCRIPTION
C      ===== ====== ========== ====================
C      .01C. SL     85/01/25.  REVISED COMMENTS.
C     ------------------------------------------------------------------
C
C     ===1=========2=========3========4=========5=========6=========7==
      IMPLICIT INTEGER(A-E,G-Z)
C
      PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=7
     +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
      DIMENSION FTRUE2(4,8)
      EQUIVALENCE (FTRUE(6),FTRUE2)
C
      DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
      CALL DATE(DTE)
      CALL TIME(TME)
      PRINT 1,LEVEL,DTE,TME
1     FORMAT('1QL32',A4,2A9)
C
100   CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
     +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
      IF(JEOF.NE.0)GOTO 900
C
      CALL WRMSTG1(FTRUE)
      IF(BUF(2).LT.MAX)GOTO 100
C
900   PRINT *,' REPORTS ',BUF(2),', EOF ',JEOF
      END
C=====================================================================
      SUBROUTINE WRMSTG1(FTRUE)
      IMPLICIT INTEGER(A-E,G-Z)
      DIMENSION FTRUE(37)
      PRINT 100,(FTRUE(I),I=1,5)
     +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
100   FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
     +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
     +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
     +1X,'U*A      ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
```

```
     +1X,'V*A      ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
     +1X,'U*Q      ',2F8.1,F8.0,F8.1,F8.0,3F8.1/
     +1X,'V*Q      ',2F8.1,F8.0,F8.1,F8.0,3F8.1)
      END
C=====================GROUP 7=========================================
      BLOCK DATA MSTG1
      IMPLICIT INTEGER(A-E,G-Z)
C
      COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
      DATA FUNITS/1., 1.,  1.,  1.,  1.
     +,0.1,  0.1,  0.1,  0.1
     +,0.1,  0.1,  0.1,  0.1
     +,1.,  1.,  1.,  1.
     +,0.1,  0.1,  0.1,  0.1
     +,2.,  2.,  2.,  2.
     +,0.1,  0.1,  0.1,  0.1
     +,0.2,  0.2,  0.2,  0.2
     +,0.2,  0.2,  0.2,  0.2/
C
      DATA FBASE/1799.,  0.,  0.,  0.,  0.
     +,-20001.,  -20001.,  -10001.,  -10001.
     +,-20001.,  -20001.,  -10001.,  -10001.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,0.,  0.,  0.,  0.
     +,-1.,  -1.,  -1.,  -1.
     +,-.5, -.5, -.5, -.5
     +,-.5, -.5, -.5, -.5/
C
      DATA BITS/8,4,14,10,12,16*16,16*4/
C
      DATA OFFSET
     +/ 16, 24, 28, 42, 52, 64, 80, 96,112,128
     +,144,160,176,192,208,224,240,256,272,288
     +,304,320,324,328,332,336,340,344,348,352
     +,356,360,364,368,372,376,380/
      END
C=====================================================================
```

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

```
C     CONVERTED BY CONVRT:  TSCON.01B                                   00100
      PROGRAM RDINV                                                     00110
C                                                                       00120
C     ****************************************************************  00130
C                                                                       00140
C             PURPOSE -       READ PACKED INVENTORIES FOR PRE-70'S OR   00150
C                             70'S DATA MADE BY PROGRAM DUPELIM         00160
C                                                                       00170
C             WRITTEN BY -    JANE HISCOX                               00180
C                                                                       00190
C     ****************************************************************  00200
C     ----------REVISION HISTORY-------------------------------------  00210
C      LEVEL AUTHOR DATE       DESCRIPTION                              00220
C      ===== ====== ========== ====================                     00230
C      .01B. SL     85/01/30.  REVISED COMMENTS; CONVERT FROM           00240
C                              TIMESHARING FORTRAN.                     00250
C     ----------------------------------------------------------------  00260
C                                                                       00270
      IMPLICIT INTEGER (A-Z)                                            00280
      CHARACTER*4 LEVEL                                                 00290
C                                                                       00300
      DIMENSION STORE (5000), CARD (50)                                 00310
C                                                                       00320
      COMMON /QC/ INVNF (14,11)                                         00330
C                                                                       00340
      DATA LEVEL /'.01B'/, NSTORE, NSID, NCD, NDS/ 5000, 24, 50, 8/     00350
      DATA RQC, CQC/ 14, 11/, BITBOX, BITYR, BITIOD, BITGT / 10, 8, 15,  00360
     +20/                                                               00370
      DATA IU, JU, OU / 1, 2, 5/                                        00380
      DATA CARD / 110, 116, 117, 118, 119, 128, 143, 150, 151, 152, 155, 00390
     +            156, 184, 185, 186, 187, 188, 189, 192, 193, 194, 195, 00400
     +            196, 197, 281, 555, 666, 849, 850, 876, 877, 878, 879, 00410
     +            880, 881, 882, 888, 889, 891, 897, 898, 899, 900, 901, 00420
     +            902, 926, 927, 928, 999, 50/                          00430
C                                                                       00440
      REWIND IU                                                         00450
      REWIND JU                                                         00460
      REWIND OU                                                         00470
C                                                                       00480
      DTE = DATE (K)                                                    00490
      TME = TIME (K)                                                    00500
      READ (JU,*,END=900) BOX                                           00510
      WRITE (5,5) BOX, LEVEL, DTE, TME                                  00520
    5 FORMAT ('1 INVENTORIES FOR BOX ',I3,T60,'BY RDINV',A,2X,2A10)     00530
C                                                                       00540
  100 BUFFER IN (IU,0) (STORE(1), STORE(NSTORE))                        00550
      IF (UNIT(IU) .LT. 0) THEN                                         00560
          OFF = 0                                                       00570
          NWORD = 1                                                     00580
          CALL GBYTE (STORE(NWORD), BOX10, OFF, BITBOX)                 00590
          IF (BOX10 .EQ. BOX) THEN                                      00600
              OFF = OFFSET (OFF,NWORD,BITBOX)                           00610
  175         CALL GBYTE (STORE(NWORD), YEAR, OFF, BITYR)               00620
              OFF = OFFSET (OFF,NWORD,BITYR)                            00630
              IF (YEAR .NE. 0) THEN                                     00640
                  YEAR = YEAR + 1799                                    00650
```

```
                  WRITE (5,200) YEAR                                          00660
  200             FORMAT (//' YEAR = ',I4,/1X,                                00670
     +                    'MO.    IN    OUT  UNCERTAIN', /1X,26('='))        00680
                  SUMI = 0                                                    00690
                  SUMO = 0                                                    00700
                  SUMD = 0                                                    00710
                  DO 225 MO = 1,12                                            00720
                     CALL GETNUM (STORE, IMO, OFF, NWORD, BITIOD)             00730
                     CALL GETNUM (STORE, OMO, OFF, NWORD, BITIOD)             00740
                     CALL GETNUM (STORE, DMO, OFF, NWORD, BITIOD)             00750
                     IF (IMO .NE. 0) WRITE (5,210) MO, IMO, OMO, DMO          00760
  210                FORMAT (1X,I2,1X,2I6,3X,I6)                              00770
                     SUMI = SUMI + IMO                                        00780
                     SUMO = SUMO + OMO                                        00790
                     SUMD = SUMD + DMO                                        00800
  225                CONTINUE                                                 00810
                     WRITE (5,250) SUMI, SUMO, SUMD                           00820
  250                FORMAT (1X,26('=')/4X,2I6,3X,I6)                         00830
C                                                                             00840
C     -----------UNPACK YEARLY TOTALS FOR SOURCE IDS                         00850
                  WRITE (5,260)                                               00860
  260             FORMAT (//' TOTALS BY SID',/                                00870
     +                    1X,'SID        IN        OUT    UNCERTAIN',/1X,   00880
     +                    36('='))                                           00890
                  SUMI = 0                                                    00900
                  SUMO = 0                                                    00910
                  SUMD = 0                                                    00920
                  DO 300 JR = 1,NSID                                          00930
                     CALL GETNUM (STORE, ISID, OFF, NWORD, BITIOD)            00940
                     CALL GETNUM (STORE, OSID, OFF, NWORD, BITIOD)            00950
                     CALL GETNUM (STORE, DSID, OFF, NWORD, BITIOD)            00960
                     IF (ISID .NE. 0) WRITE (5,275) JR, ISID, OSID, DSID      00970
  275                FORMAT (1X,I3,3(3X,I7))                                  00980
                     SUMI = SUMI + ISID                                       00990
                     SUMO = SUMO + OSID                                       01000
                     SUMD = SUMD + DSID                                       01010
  300                CONTINUE                                                 01020
                     WRITE (5,325) SUMI, SUMO, SUMD                           01030
  325                FORMAT (1X,36('='),/4X,3(3X,I7))                         01040
                     GO TO 175                                                01050
               ENDIF                                                          01060
C                                                                             01070
C     -----------UNPACK GRAND TOTALS BY SID                                  01080
               WRITE (5,350) BOX10                                            01090
  350          FORMAT ('1 GRAND TOTALS FOR BOX ',I3,//                        01100
     +                 1X,' SID       IN       OUT  UNCERTAIN',/1X,         01110
     +                 36('='))                                              01120
               SUMI = 0                                                       01130
               SUMO = 0                                                       01140
               SUMD = 0                                                       01150
               DO 400 JR = 1,NSID                                             01160
                  CALL GETNUM (STORE, ISID, OFF, NWORD, BITGT)                01170
                  CALL GETNUM (STORE, OSID, OFF, NWORD, BITGT)                01180
                  CALL GETNUM (STORE, DSID, OFF, NWORD, BITGT)                01190
                  IF (ISID .NE. 0) WRITE (5,275) JR, ISID, OSID, DSID         01200
                  SUMI = SUMI + ISID                                          01210
```

```
                SUMO = SUMO + OSID                                              01220
                SUMD = SUMD + DSID                                              01230
  400           CONTINUE                                                        01240
                WRITE (5,325) SUMI, SUMO, SUMD                                  01250
C                                                                               01260
C     -----------UNPACK GRAND TOTALS BY CARD DECK                               01270
                WRITE (5,500)                                                   01280
  500           FORMAT (///,1X,' CD        IN       OUT  UNCERTAIN',/1X,       01290
     +                 36('='))                                                 01300
                SUMI = 0                                                        01310
                SUMO = 0                                                        01320
                SUMD = 0                                                        01330
                DO 600 JR = 1,NCD                                               01340
                   CALL GETNUM (STORE, ICD, OFF, NWORD, BITGT)                  01350
                   CALL GETNUM (STORE, OCD, OFF, NWORD, BITGT)                  01360
                   CALL GETNUM (STORE, DCD, OFF, NWORD, BITGT)                  01370
                   IF (ICD .NE. 0) WRITE (5,275) CARD(JR), ICD, OCD, DCD        01380
                   SUMI = SUMI + ICD                                            01390
                   SUMO = SUMO + OCD                                            01400
                   SUMD = SUMD + DCD                                            01410
  600           CONTINUE                                                        01420
                WRITE (5,325) SUMI, SUMO, SUMD                                  01430
C                                                                               01440
C     -----------UNPACK GRAND TOTALS                                           01450
                WRITE (5,625)                                                   01460
  625           FORMAT (///' GRAND TOTALS')                                     01470
                CALL GETNUM (STORE, IGT, OFF, NWORD, BITGT)                     01480
                CALL GETNUM (STORE, OGT, OFF, NWORD, BITGT)                     01490
                CALL GETNUM (STORE, DGT, OFF, NWORD, BITGT)                     01500
                WRITE (5,650) IGT, OGT, DGT                                     01510
  650           FORMAT (/' TOTAL IN = ',I7,',  TOTAL OUT = ',I7,               01520
     +                 ',  NUMBER OF UNCERTAIN IN OUT = ',I7)                  01530
C                                                                               01540
C     -----------UNPACK TOTALS BY DS                                           01550
                WRITE (5,675)                                                   01560
  675           FORMAT (///' TOTALS BY DUPLICATE STATUS',//5X,                  01570
     +                 ' DS  TOTAL',/5X,12('='))                               01580
                SUMDS = 0                                                       01590
                DO 700 JR = 1,NDS                                               01600
                   CALL GETNUM (STORE, ODS, OFF, NWORD, BITGT)                  01610
                   J = JR - 1                                                   01620
                   WRITE (5,685) J, ODS                                         01630
  685              FORMAT (5X,I3,I7)                                            01640
                   SUMDS = SUMDS + ODS                                          01650
  700           CONTINUE                                                        01660
                WRITE (5,725) SUMDS                                             01670
  725           FORMAT (5X,12('='),/8X,I7)                                      01680
C                                                                               01690
C     -----------UNPACK QC INVENTORIES                                         01700
                DO 800 JC = 1,CQC                                               01710
                   DO 775 JR = 1,RQC                                            01720
                      CALL GETNUM (STORE, INVNF(JR,JC), OFF, NWORD, BITGT)      01730
  775              CONTINUE                                                     01740
  800           CONTINUE                                                        01750
C               CALL PRINVN (BOX10)                                             01760
                GO TO 900                                                       01770
```

```
          ENDIF                                                          01780
          GO TO 100                                                      01790
      ENDIF                                                              01800
  900 REWIND IU                                                          01810
      REWIND JU                                                          01820
      REWIND OU                                                          01830
      END                                                                01840
C                                                                        01850
C     ******************************************************************* 01860
C                                                                        01870
      SUBROUTINE GETNUM (STORE, NUM, OFF, NWORD, BITS)                   01880
C                                                                        01890
C     ----------UNPACK NUMBER, UPDATE OFFSET.  IF THE UNPACKED NUMBER    01900
C               IS THE MAXIMUM SIZE FOR NUMBER OF BITS, UNPACK THE NEXT  01910
C               NUMBER AND SUM THEM.                                     01920
C                   STORE    - ARRAY TO UNPACK NUMBER FROM               01930
C                   NUM      - RESULTANT NUMBER                          01940
C                   OFF      - OFFSET                                    01950
C                   NWORD    - WORD OF ARRAY STORE TO UNPACK FROM        01960
C                   BITS     - NUMBER OF BITS TO UNPACK FROM STORE       01970
C                                                                        01980
      IMPLICIT INTEGER (A-Z)                                             01990
C                                                                        02000
      DIMENSION STORE (*)                                                02010
C                                                                        02020
      NUM = 0                                                            02030
  100 CALL GBYTE (STORE(NWORD), N, OFF, BITS)                            02040
      OFF = OFFSET (OFF, NWORD, BITS)                                    02050
      NUM = NUM + N                                                      02060
      IF (N .GE. (2**BITS - 1)) GO TO 100                                02070
      END                                                                02080
C                                                                        02090
C     ******************************************************************* 02100
C                                                                        02110
      INTEGER FUNCTION OFFSET (OFF, NWORD, BITS)                         02120
C                                                                        02130
C     ----------UPDATE OFFSET AND NWORD BY BITS                          02140
C                                                                        02150
      IMPLICIT INTEGER (A-Z)                                             02160
      DATA WRDSIZ / 60/                                                  02170
C                                                                        02180
      OFFSET = OFF + BITS                                                02190
      IF (OFFSET .GE. WRDSIZ) THEN                                       02200
          OFFSET = OFFSET - WRDSIZ                                       02210
          NWORD = NWORD + 1                                              02220
      ENDIF                                                              02230
      END                                                                02240
C                                                                        02250
C     ******************************************************************* 02260
C                                                                        02270
      SUBROUTINE PRINVN (BOX10)                                          02280
C                                                                        02290
C     ----------PRINT QC INVENTORIES                                     02300
C                                                                        02310
      IMPLICIT INTEGER (A-Z)                                             02320
      CHARACTER FLAG (14)*8                                              02330
```

```
C                                                                       02340
      COMMON /QC/ INVNF (14,11)                                         02350
C                                                                       02360
      DATA FLAG /'SHIP POS','WIND    ','VIS     ','PRES WX ','PAST WX ',02370
     +           'PRESSURE','DRY BULB','WET BULB','DEW PT  ','SEA TEMP',02380
     +           'CLOUDS  ','WAVES   ','SWELLS  ','P TEND  '/         02390
C                                                                       02400
      WRITE (5,10) BOX10                                                02410
   10 FORMAT (///,' QUALITY CONTROL FLAGS, BOX10 = ',I3,                02420
     +          /1X,'FLAG/VALUE',3X,'MISSING',7X,'R',9X,'A',9X,'B',9X, 02430
     +          'J',9X,'K',9X,'L',9X,'M',9X,'N',9X,'Q',9X,'S',5X,      02440
     +          'TOTAL')                                               02450
      DO 230 JR = 1,14                                                  02460
         TOTAL = 0                                                      02470
         DO 220 JC = 1,11                                               02480
            TOTAL = TOTAL + INVNF(JR,JC)                                02490
  220 CONTINUE                                                          02500
         WRITE (5,225) FLAG(JR),(INVNF(JR,JC),JC=1,11),TOTAL            02510
  225 FORMAT (1X,A,12I10)                                               02520
  230 CONTINUE                                                          02530
      END                                                               02540
```

```
C     CONVERTED BY CONVRT:  TSCON.01B                                    00100
      SUBROUTINE READER(UNIT,TARGET)                                     00110
C     ----------READ LANDLOCKED BOX2 MAP INTO @TARGET(16202)            00120
C               FROM INTEGER &UNIT.                                      00130
C                    1H. = LAND                                          00140
C                    1H* = COASTAL                                       00150
C                    1H  = SEA                                           00160
C     ----------REVISION HISTORY---------------------------------------- 00170
C      LEVEL AUTHOR DATE       DESCRIPTION                               00180
C      ===== ====== ========== ==================                        00190
C      .01A. SDW    85/02/15.  ORIGINAL VERSION TAKEN FROM LLLIBS.01J.   00200
C      .01B. SL     85/02/15.  REPLACE ALL R1 FORMAT DESCRIPTORS WITH     00210
C                              A1.  REMOVE CONVERT TO INTEGER ENTRY.      00220
C                              REMOVE ALL END= FROM READ STATEMENTS.      00230
C                              REVISED COMMENTS.  CONVERT FROM            00240
C                              TIMESHARING FORTRAN.                       00250
C     ------------------------------------------------------------------ 00260
      IMPLICIT INTEGER(A-E,G-Z)                                          00270
      DIMENSION TARGET(16202)                                            00280
C     ----------READ, @TARGET WILL REMAIN IN A1 WITH NO CONVERSION       00290
      READ(UNIT,100) TARGET(1)                                           00300
  100 FORMAT(///,6X,A1)                                                  00310
      DO 300 KLAT=1,90                                                   00320
        KLON1=(KLAT-1)*180+2                                             00330
        KLON2=KLON1+89                                                   00340
        READ(UNIT,200) (TARGET(I),I=KLON1,KLON2)                         00350
  200   FORMAT(6X,90A1)                                                  00360
  300 CONTINUE                                                           00370
      READ(UNIT,350)                                                     00380
  350 FORMAT(3(/))                                                       00390
      DO 500 KLAT=1,90                                                   00400
        KLON1=(KLAT-1)*180+92                                            00410
        KLON2=KLON1+89                                                   00420
        READ(UNIT,200) (TARGET(I),I=KLON1,KLON2)                         00430
  500 CONTINUE                                                           00440
      READ(UNIT,600) TARGET(16202)                                       00450
  600 FORMAT(95X,A1)                                                     00460
      END                                                                00470
```