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>
<thead>
<tr>
<th>Filename</th>
<th>Level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOXLIB</td>
<td>.01J</td>
<td>tools for working with 2°, 4°, and 10° boxes, or Marsden Squares</td>
</tr>
<tr>
<td>Q19</td>
<td>.01G</td>
<td>read and print MSU.2</td>
</tr>
<tr>
<td>Q112</td>
<td>.01D</td>
<td>read and print CMR.4</td>
</tr>
<tr>
<td>Q121</td>
<td>.01D</td>
<td>read and print MSUG.1 group 1</td>
</tr>
<tr>
<td>Q122</td>
<td>.01D</td>
<td>read and print MSUG.1 group 2</td>
</tr>
<tr>
<td>Q124</td>
<td>.01C</td>
<td>read and print DSU.2</td>
</tr>
<tr>
<td>QL14</td>
<td>.01C</td>
<td>read and print MST.3</td>
</tr>
<tr>
<td>QL16</td>
<td>.01C</td>
<td>read and print TRP.1</td>
</tr>
<tr>
<td>QL21</td>
<td>.01C</td>
<td>read and print CMR.5</td>
</tr>
<tr>
<td>QL28</td>
<td>.01C</td>
<td>read and print MSTG.1 group 3</td>
</tr>
<tr>
<td>QL29</td>
<td>.01C</td>
<td>read and print MSTG.1 group 4</td>
</tr>
<tr>
<td>QL30</td>
<td>.01C</td>
<td>read and print MSTG.1 group 5</td>
</tr>
<tr>
<td>QL31</td>
<td>.01C</td>
<td>read and print MSTG.1 group 6</td>
</tr>
<tr>
<td>QL32</td>
<td>.01C</td>
<td>read and print MSTG.1 group 7</td>
</tr>
<tr>
<td>RDI NV</td>
<td>.01B</td>
<td>read and print INV.3</td>
</tr>
<tr>
<td>READER</td>
<td>.01B</td>
<td>read landlocked file LLN2F1</td>
</tr>
<tr>
<td>LLN2F1</td>
<td>n/a</td>
<td>landlocked file</td>
</tr>
</tbody>
</table>

* 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>
<thead>
<tr>
<th>Incompatibility</th>
<th>BOXLIB</th>
<th>Q19-QL32</th>
<th>RDINV</th>
<th>READER</th>
<th>LLN2F1</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN 66</td>
<td>X</td>
<td>X^a</td>
<td>X</td>
<td>n/a</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>7-char variables</td>
<td></td>
<td>X^b</td>
<td></td>
<td>n/a</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>DATE</td>
<td>X</td>
<td></td>
<td>X</td>
<td>n/a</td>
<td></td>
<td>this section</td>
</tr>
<tr>
<td>TIME</td>
<td></td>
<td>X</td>
<td>X</td>
<td>n/a</td>
<td></td>
<td>this section</td>
</tr>
<tr>
<td>BPW (bits/word)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>n/a</td>
<td></td>
<td>this section</td>
</tr>
<tr>
<td>BUFFER IN</td>
<td></td>
<td></td>
<td></td>
<td>n/a</td>
<td></td>
<td>sec. 1</td>
</tr>
<tr>
<td>UNIT</td>
<td></td>
<td>X</td>
<td></td>
<td>n/a</td>
<td></td>
<td>sec. 1</td>
</tr>
<tr>
<td>LENGTH</td>
<td>X</td>
<td></td>
<td></td>
<td>n/a</td>
<td></td>
<td>sec. 1</td>
</tr>
<tr>
<td>RPTIN</td>
<td></td>
<td>X</td>
<td></td>
<td>n/a</td>
<td></td>
<td>sec. 1</td>
</tr>
<tr>
<td>GBYTES</td>
<td>X</td>
<td></td>
<td></td>
<td>n/a</td>
<td></td>
<td>sec. 2</td>
</tr>
<tr>
<td>DEC computer</td>
<td>X</td>
<td>X</td>
<td></td>
<td>n/a^e</td>
<td></td>
<td>sec. 3</td>
</tr>
</tbody>
</table>

^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

\[ \text{BUFFER IN(}LUN,M\text{) (}K(1),K(N)\text{)} \]

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

\[ \text{JEOF=}\text{UNIT(LUN)} \]

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 "An" format where \( n \) is the number of characters per word. For example, on a 32-bit IBM machine with 8-bit characters,

\[
\text{INTEGER K(1800)} \\
100 \ \text{READ}(1,200) \ K \\
200 \ \text{FORMAT}(1800A4)
\]

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

* Programs Q19-QL32 have this dimension set to the integer parameter \( \text{DIM BUF} = (1006 + 64-1)/\text{BPW} + 1 \) for compatibility with RPTIN. Since RPTIN is not called in the default implementation, \( \text{DIM BUF} \) 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 ≤ Q < word size, and 1 ≤ B ≤ 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.
C

PROGRAM TEST
CHARACTER*10 LEVEL*6,DTE,TME
INTEGER UNIT
DATA LEVEL/' .01J. '/
CALL DATE(DTE)
CALL TIME(TME)
PRINT 1,LEVEL,DTE,TME
1 FORMAT(1B1XPORT',3A)
WRITE(UNIT,1) LEVEL,DTE,TME
RETURN
END

C

*******************************************************************************
C
-----BXPOR\, SOURCE CODE FOR BOXLIB
C
A LIBRARY OF TOOLS FOR USING BOXES AND OTHER GLOBAL
C
GRID SYSTEMS, E.G. MARSDEN SQUARES. THE BOX SYSTEMS ARE:
C
GENERIC NAME SPECIFIC NAME POLAR BOXES X-ORIGIN
------------- ------- --------- -------
BOX2 BX16202 YES \OE
BOX4 BX4052 YES \OE
BOX10 BX648 NO \OE

==1==2==3==4==5==6==7==

C

---------REVISION HISTORY-----------------------------
C
LEVEL AUTHOR DATE DESCRIPTION
----- ---- ------- ------------------
.01A. --- 83/07/20. ORIGINAL VERSION TAKEN QLIBS.01I VIA F45
.01B. SDW 83/07/21. UPDATES BOX10 TOOLS TO CURRENT SYSTEM
.01C. SDW 84/05/02. FIX ERROR IN <XYBQ>, COMMENT OUT <XYMSQ>,
                 AND ADD <B1026>.
.01D. TSP 84/10/05. FIXED <B1OXYO> TO ADJUST FOR 30 DEGREE
                 SHIFT OF B10 SYSTEM
.01E. TSP 84/10/08. FIXED ERRORS IN <MSQ8B10>
.01F. TSP 84/10/08. FIXED <XYMSQ> AND <MSQXYO>
.01G. TSP 84/10/09. DELETED <B25> AND <B52>, TRIMMED ALL
                 LINES TO 72 CHARACTERS MAXIMUM
.01H. TSP 84/10/09. DELETED <B5XYO>, <MSQ5>, AND <XYB5>
.01I. TSP 84/10/10. CHANGED NAMES OF SOURCE AND
                 OBJECT CODE.
.01J. TSP 84/10/15. DELETED BOX5 AND AUTHOR COMMENT LINES.

==1==2==3==4==5==6==7==

C

INTEGER FUNCTION B1OMSQ(MSQ)
C
EQUIVALENT -1 IF ILLEGAL MSQ ELSE EQUALS EQUIVALENT B10
IMP\IMPLICIT INTEGER(A-Z)
IF(MSQ.GE.1.AND.MSQ.LE.288)THEN
  SQR=MSQ+35
ELSE IF(MSQ.GE.300.AND.MSQ.LE.623)THEN
  SQR=-1*(MSQ-300)
ELSE IF(MSQ.GE.901.AND.MSQ.LE.936)THEN
  SQR=MSQ-577
ELSE
  GOTO 900
ENDIF
B1OMSQ=(9-SQR/36)*36+(71-MOD(IABS(SQR),36))
+(-(71-MOD(IABS(SQR),36))/39*36 -2

- H6 -
RETURN
900 BIOMSQ=-1
RETURN
END
C
====1======2======3======4======5======6======7=====
LOGICAL FUNCTION BI026(B2,B26,B10)
C
--------FALSE IF 1<QB10<648, ELSE TRUE SUCH THAT QB2 CONTAINS
C
THE 25 BOX2 CONTAINED BY BOX10 QB10 IN NUMERICAL ORDER,
C
AND QB26 CONTAINS ZERO OR THE 26TH BOX2 FOR THE POLAR
C
BOX10.
C
IMPLICIT INTEGER(A-Z)
LOGICAL XYB10,B2XYO
DIMENSION B2(25)
JB=B26=0
BI026=.FALSE.
IF(.NOT.XYB10(X1,Y2,B10)) RETURN
X2=X1+80
Y1=Y2+80
DO 500 Y=Y1,Y2,-20
   DO 500 X=X1,X2,20
      IF(.NOT.B2XYO(X,Y,BOX2)) RETURN
         JB=JB+1
   B2(JB)=BOX2
500 CONTINUE
IF(B10.EQ.1) B26=1
IF(B10.EQ.648) B26=16202
BI026=.TRUE.
RETURN
END
C
====1======2======3======4======5======6======7=====
LOGICAL FUNCTION B10XYO(X,Y,B10)
C
--------PERFORM <BQXYO> ON 10 DEGREE BOX CORNER QX,QY
C
IMPLICIT INTEGER(A-E,G-Z)
LOGICAL BQXYO
DATA Q/100/,XDIM/36/,Y1/800/,YMVE/8/,X2/3500/
C
-- SHIFT LATITUDE X 30 DEGREES WEST TO COMPUTE USING BQXYO
IF (X .GE. 300) THEN
   X=X-300
ELSE
   X=X+3300
ENDIF
B10XYO=BQXYO(XS,Y,B10,Q,XDIM,Y1,YMVE,X2)
C
-- SUBTRACT 1 FROM BOX # TO ADJUST FOR LACK OF NORTH POLAR BOX
B10=B10-1
RETURN
END
C
====1======2======3======4======5======6======7=====
F45V1PO
LOGICAL FUNCTION B2XYO(X,Y,B2)
C
--------PERFORM <BQXYO> ON 2 DEGREE BOX CORNER QX,QY
C
IMPLICIT INTEGER(A-E,G-Z)
LOGICAL BQXYO
DATA Q/20/,XDIM/180/,Y1/880/,YMVE/44/,X2/3580/
B2XYO=BQXYO(X,Y,B2,Q,XDIM,Y1,YMVE,X2)
RETURN
END
***1==========2========3==========4========5==========6========7==

*F45V1PO*

LOGICAL FUNCTION B4XYO(X,Y,B4)

---- FALSE IF QX,QY ARE NOT THE LOWER-LEFT (SW) CORNER OF A

Q/10 DEGREE BOX IN 10THS DEGREE N,-S,E.

ELSE TRUE RETURNING THE BOX NUMBER QB4

WHERE QXDIM IS THE NUMBER OF BOXES PER LAT ZONE

QY1 IS 900-QQ

QX2 IS THE LARGEST X

WARNING - DO NOT USE THIS FUNCTION FOR THE POLAR BOXES.

<B4XYO> CANNOT RECOGNIZE (0,900) AS THE SOUTHWEST

CORNER OF THE NORTH POLAR BOX, AND ALL BOXES IN THE

-85 TO -90 DEGREE LATITUDE BAND HAVE (QX,QY)=(0,900)

AS THEIR SOUTHWEST CORNER. THE <B4XYO> CANNOT TELL

WHICH BOX IS THE SOUTH POLAR BOX WHEN GIVEN (0,-900).

<B4XYO> RETURNS .FALSE. FOR NORTH POLAR BOX.

RETURNS .TRUE. FOR SOUTH POLAR BOX; BUT

THE RETURNED BOX IS NOT THE SOUTH POLAR

BOX.

IMPLICIT INTEGER(A-E,G-Z)
DATA Q/40/,QDIM/90/,Y1/860/,X2/3560/

IF(Q/10.EQ.0.AND.MOD(900-Q).EQ.0.AND.

+ (X.GE.0.AND.X.LE.X2).AND.

+ (Y.GE.-900.AND.Y.LE.Y1)) GOTO 200

B4XYO=.FALSE.

RETURN

200 B4=((900-Y)/Q-1)*XDIM*X/Q+2

B4XYO=.TRUE.

RETURN

END

***1==========2========3==========4========5==========6========7==

*F45V1PO*

LOGICAL FUNCTION BQXYO(X,Y,BQ,Q,XDIM,Y1,YMOVE,X2)

---- FALSE IF QX,QY ARE NOT THE LOWER-LEFT (SW) CORNER OF A Q/10

DEGREE BOX IN 10THS DEGREE +N,-S,E; EXCLUDING POLAR BOXES

ELSE TRUE RETURNING THE BOX NUMBER QBQ

WHERE QXDIM IS THE NUMBER OF BOXES PER LAT ZONE

QY1 IS 900-QQ

QYM搬 IS (900/QQ)-1

QX2 IS THE LARGEST X

WARNING - DO NOT USE THIS FUNCTION FOR THE POLAR BOXES.

<BQXYO> CANNOT RECOGNIZE (0,900) AS THE SOUTHWEST

CORNER OF THE NORTH POLAR BOX, AND ALL BOXES IN THE

-85 TO -90 DEGREE LATITUDE BAND HAVE (QX,QY)=(0,900)

AS THEIR SOUTHWEST CORNER. THE <BQXYO> CANNOT TELL

WHICH BOX IS THE SOUTH POLAR BOX WHEN GIVEN (0,-900).

<BQXYO> RETURNS .FALSE. FOR NORTH POLAR BOX.

RETURNS .TRUE. FOR SOUTH POLAR BOX; BUT

THE RETURNED BOX IS NOT THE SOUTH POLAR

BOX.

- H8 -
IMPLICIT INTEGER(A-E,G-Z)
IF(MOD(X,Q).EQ.0.AND.MOD(Y,Q).EQ.0.AND.  
+ (X.GE.0.AND.X.LE.X2)) AND.  
+ (Y.GE.-900.AND.Y.LE.Y1)) GOTO 200
BQXYO=.FALSE.
RETURN
200 BQ=(YMOVE-Y/Q)*XDIM+X/Q+2
BQXYO=.TRUE.
RETURN
C ** THIS PROGRAM VALID ON FTN4 AND FTN5 **
END
C ==1==2==3==4==5==6==7==
INTEGER FUNCTION MSQB10(B10)
C ----------EQUALS -1 IF ILLEGAL B10, ELSE EQUALS EQUIVALENT MSQ
IMPLICIT INTEGER(A-E,G-Z)
MSQB10=-1
M=MOD(B10,36)
IF (M .EQ. 0) M=36
IF (B10 .GE. 1 .AND. B10 .LE. 33) THEN
  MSQB10 = 934-B10
ELSE
  MSQB10 = 970-B10
ENDIF
IF (B10 .GE. 37 .AND. B10 .LE. 324) THEN
  IF (M .GE. 1 .AND. M .LE. 33) THEN
    MSQB10 = 322-B10
  ELSE
    MSQB10 = 358-B10
  ENDIF
ENDIF
IF (B10 .GE. 325 .AND. B10 .LE. 648) THEN
  IF (M .GE. 1 .AND. M .LE. 33) THEN
    MSQB10 = 333-M+((AINT(B10/36.0)-9)*36)
  ELSE IF (M .EQ. 34 .OR. M .EQ. 35) THEN
    MSQB10 = 369-M+((AINT(B10/36.0)-9)*36)
  ELSE IF (M .EQ. 36) THEN
    MSQB10 = 333+(AINT(B10/36.0)-10)*36
  ENDIF
ENDIF
RETURN
END
C ==1==2==3==4==5==6==7==
C *F45V1P0*
LOGICAL FUNCTION MSQXYO(X,Y,MSQ)
C ----------RETURNS MSQ BOX# 0MSQ GIVEN 10 DEGREE BOX CORNER QX, QY
C RETURNS FALSE IF QX,QY IS NOT THE CORNER OF A 10 DEGREE
C BOX.
C C <MSQXYO> USES <BQXYO> - SEE WARNING BELOW.
C C WARNING - DO NOT USE THIS FUNCTION FOR THE POLAR BOXES.
C C <BQXYO> CANNOT RECOGNIZE (0,900) AS THE SOUTHWEST
C CORNER OF THE NORTH POLAR BOX, AND ALL BOXES IN THE
C -85 TO -90 DEGREE LATITUDE BAND HAVE (QX,QY)=(0,-900)
C AS THEIR SOUTHWEST CORNER.  THUS <BQXYO> CANNOT TELL
C WHICH BOX IS THE SOUTH POLAR BOX WHEN GIVEN (0,-900).
C
H9
C
C   \langle BQXYO\rangle  RETURNS .FALSE. FOR NORTH POLAR BOX.
C   RETURNS .TRUE. FOR SOUTH POLAR BOX; BUT
C   THE RETURNED BOX IS NOT THE SOUTH POLAR
C   BOX.
C
C   IMPLICIT INTEGER(A-E,G-Z)
C   LOGICAL BQXYO
C   -- SHIFT LATITUDE X 30 DEGREES WEST TO COMPUTE USING BQXYO
C   IF (X .GE. 300) THEN
C      XS=X-300
C   ELSE
C      XS=X+300
C   ENDIF
C   DATA Q/100/,XDIM/36/,Y1/800/,YMOVE/8/,X2/3500/
C   MSQXYO=BQXYO(XS,Y,BQ,Q,XDIM,Y1,YMOVE,X2)
C   -- SUBTRACT 1 FROM BOX # TO ADJUST FOR LACK OF POLAR BOX AND
C   RECALCULATE THE EQUIVALENT MARSDEN SQUARE
C   MSQ=MSQB10(BQ-1)
C   RETURN
C   END
C
C   ==1=2==3==4==5==6==7==
C   INTEGER FUNCTION QCDCXY(X,Y)
C   --RETURNS -1 UNLESS 900<Y<-900, 3599<X<0, QX>1800 (10THS E)
C   RETURNS THE NCDC QUADRANT 1=NW,2=NE,3=SW,4=SE OTHERWISE
C   IMPLICIT INTEGER(A-E,G-Z)
C   IF(Y.LT.900.AND.Y.GT.-900.AND.X.LT.3599.AND.X.GT.0.AND.X.NE.1800)
C      QCDCXY=1
C      IF(X.LT.1800) QCDCXY=QCDCXY+1
C      IF(Y.LT.0) QCDCXY=QCDCXY+2
C   ELSE
C      QCDCXY=-1
C   ENDIF
C   RETURN
C   END
C
C   ==1==2==3==4==5==6==7==
C   LOGICAL FUNCTION XYB10(X,Y,B10)
C   --PERFORM \langle XYBQ\rangle ON A 10 DEGREE BOX QB10
C   IMPLICIT INTEGER(A-E,G-Z)
C   LOGICAL XYBQ
C   DATA Q/100/,LAST/648/,XDIM/36/,Y1/800/,POLE/1/,XMOVE/300/
C   XYB10=XYBQ(X,Y,B10,Q,LAST,XDIM,Y1,POLE,XMOVE)
C   RETURN
C   END
C
C   ==1==2==3==4==5==6==7==
C   \(*F45V1P0*\)
C   LOGICAL FUNCTION XYB2(X,Y,B2)
C   --PERFORM \langle XYBQ\rangle ON A 2 DEGREE BOX QB2
C   IMPLICIT INTEGER(A-E,G-Z)
C   LOGICAL XYBQ
C   DATA Q/20/,LAST/16202/,XDIM/180/,Y1/880/,POLE/2/,XMOVE/0/
C   XYB2=XYBQ(X,Y,B2,Q,LAST,XDIM,Y1,POLE,XMOVE)
C   RETURN
C   END
C
C   ==1==2==3==4==5==6==7==

- H10 -
C *F45V1PO*
LOGICAL FUNCTION XYB4(X,Y,B4)
C -------------PERFORM <XYBQ> ON A 4 DEGREE BOX @B4
IMPLICIT INTEGER(A-E,G-Z)
LOGICAL XYBQ
DATA Q/40/,.LAST/4052/,.XDIM/90/,.Y1/860/,.POLE/2/,.XMOVE/0/
XYB4=XYBQ(X,Y,B4,.LAST,.XDIM,.Y1,.POLE,.XMOVE)
RETURN
END
C ===1============2============3============4============5============6============7==
C *F45V1PO*
LOGICAL FUNCTION XYBQ(X,Y,BQ,.LAST,.XDIM,.Y1,.POLE,.XMOVE)
C -------FALSE IF 1>BQ>0-LAST, ELSE TRUE SUCH THAT 0X,0Y ARE THE
C LAT, LON IN 10THS DEGREE +N,-S,E OF LOWER-LEFT (SW) CORNER
C OF 0Q/10 DEGREE BOX @BQ; POLAR 0X ARE SET TO 0
C WHERE 0-LAST IS THE LAST BOX NUMBER
C 0XDIM IS THE NUMBER OF BOXES PER LAT ZONE
C 0Y1 IS 900-0Q
C 0POLE IS 1 IF O POLAR BOXES, 2 IF 2 POLAR BOXES
C 0XMOVE IS THE X-ORIGIN
IMPLICIT INTEGER(A-E,G-Z)
XYBQ=.FALSE.
IF(BQ.LT.1.OR.BQ.GT.LAST) RETURN
IF(POLE.EQ.1) GOTO 200
IF(BQ.NE.1) GOTO 100
X=0
Y= 900
GOTO 900
100 IF(BQ.NE.LAST) GOTO 200
X=0
Y= 900
GOTO 900
200 CONTINUE
X=MOD(BQ-POLE,.XDIM)*Q*XMOVE
IF(X.GE.3600) X=X-3600
Y=Y1-(BQ-POLE)/.XDIM*Q
900 XYBQ=.TRUE.
RETURN
C ** THIS PROGRAM VALID ON FTN4 AND FTN5 **
END
C ===1============2============3============4============5============6============7==
C *F45V1PO*
LOGICAL FUNCTION XYMSQ(X,Y,MSQ)
C ------------- PERFORM <B1OMSQ> TO CONVERT @MSQ TO @B10, THEN USES
C <XYBQ> TO FIND LAT. AND LONG. OF EQUIVALENT @B10
IMPLICIT INTEGER(A-E,G-Z)
LOGICAL XYBQ
B10 = B1OMSQ(MSQ)
DATA Q/100/,.LAST/648/,.XDIM/36/,.Y1/800/,.POLE/1/,.XMOVE/300/
XYMSQ=XYBQ(X,Y,B10,.LAST,.XDIM,.Y1,.POLE,.XMOVE)
RETURN
END
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 CORES 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
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---
C
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=100,RPTOFF=1,FMISS=-99999,INDEXCK=5,BPR=1600,ID=0
C +,BPW=60,DIM BUF=(1006+64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=117)
C
C COMMON /MSU2/FUNITS(117),FBASE(117),BITS(117),OFFSET(117)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C------2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(8,14)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01G/,,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C 1 FORMA'T(1QI9',A4,2A9)
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C PRINT 300,FTRUE
C 300 FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0,' BOX10 ',F4.0
C +,' CHECKSUM ',F6.0/
C +,'X',S',7X,'A',7X,'W',7X,'U',7X,'V',7X,'P',7X,'C',7X,'Q'/'
C +,'Y',8F8.1/
C +,'H',8F8.1/
C +,'X',8F8.2/
C +,'Y',8F8.2/
C +,'N',8F8.0/
C +,'M',6F8.2,F8.1,F8.2/
C +,'S',6F8.2,F8.1,F8.2/
C +,'O',6F8.2,F8.1,F8.2/
C +,'1',6F8.2,F8.1,F8.2/
C +,'2',6F8.2,F8.1,F8.2/
C +,'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)G0T0 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
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 GETRP(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
C INPUT
C TAPE - RPTIN/RCDDIN UNIT
C FMISS - MISSING VALUE

-H13-
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
C IMPPLICIT 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-----BYTE AND CONVERT TO TRUE
200  CK=ID
  DO 230 I=1,DIM UN
  CALL QBYTE(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

-H14-
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
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
100 BUF(1)=0
    BUF(5)=LENGTH(TAPE)*BPW
   IF(JEOF.EQ.1)RETURN
   BUF(3)=BUF(3)+1
C
C-----GBTYE
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, QBYTE/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=ечение3=ечение4=ечение5=ечение6=ечение7===
C
C--------REVISION HISTORY-------------------------------------------
C LEVEL AUTHOR DATE DESCRIPTION
C ===1=ечение2=ечение3=щение4=щение5=щение6=щение7===
C .01D.SL 85/01/25. REVISED COMMENTS.
C
C ===1=ечение2=ечение3=щение4=щение5=щение6=щение7===
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=300,RPTOFF=1,FMISS=-999.9,INDEXCK=30,BPR=192,ID=0
C +,BPW=60,DIM BUF=(100*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=30)
C
C COMMON /CMR4/FIELD(30),FTRUEL(30),FTRUEU(30),FUNITS(30)
C +,FBASE(30),BITS(30),OFFSET(30)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C DATA LEVEL/4H.01D/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C 1 FORMAT('1QI12',A4,2A9)
C
C 100 CALL GETRP(T1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C PRINT 300,(FIELD(I),FTRUE(I),I=1,DIM UN)
C 300 FORMAT(6(1X,A5,F7.1))
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *,', REPORTS ',BUF(2),', ', EOF ',JEOF
C END
C
C-------------------------------
C BLOCK DATA CMR4
C IMPLICIT INTEGER(A-E,G-Z)
C
C COMMON /CMR4/FIELD(30),FTRUEL(30),FTRUEU(30),FUNITS(30)
C +,FBASE(30),BITS(30),OFFSET(30)
C
C DATA FIELD/
C +8HBOX10 ,8MONTH ,8HBOX2 ,8YEAR ,8HDAY ,
C +8HOUR ,8X ,8Y ,8S ,8BI ,
C +8HA ,8DP ,8THI ,8W ,8WI ,
C +8HU ,8V ,8HDI ,8P ,8HC ,
C - H16 -
DATA FTRUEL/
+3*1.,1800.,1.,3*0.,-5.,0.,-88.,4*0.,2*-102.2,0.,870.,11*0. /

DATA FTRUEU/
+102.2,5.,1074.6,2*9.,2*10.,1.,11*0.,7.,99.,999.,62. /

DATA FUNITS/
+6*1.,3*1.,1.,2*1.,1.,1.,1.,2*,1.,1.,1.,1,11*1. /

DATA FBASE/
+3*0,1799,0,3*-1,-51,-1,-881,4*-1,2*-1023,-1,8699,10*-1,0/

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/

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

=============================================

----- SEE QI9 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----
PROGRAM QI21
C-----READ AND PRINT WSUG1 GROUP1
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, QBYTE/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
C==1=-------2=-------3=-------4=-------5=-------6=-------7==
C
C-------REVISION HISTORY--------------------------------------
C LEVEL AUTHOR DATE DESCRIPTION
C==1=-------2=-------3=-------4=-------5=-------6=-------7==
C .01D. SL 85/01/25. REVISED COMMENTS.
C
C==1=-------2=-------3=-------4=-------5=-------6=-------7==
C
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=1
C +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
C COMMON /MSUG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(4,8)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01D/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C 1 FORMAT('IQI21',A4,2A9)
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,IDX
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C CALL WRMSUG1(FTRUE)
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *,', REPORTS ',BUF(2),', ',EDF ',',JEVF
C END
C
C=====================================================================
C SUBROUTINE WRMSUG1(FTRUE)
C IMPLICIT INTEGER(A-E,G-Z)
C DIMENSION FTRUE(37)
C PRINT 100,(FTRUE(I),I=1,5)
C +,((FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
C 100 FORMAT('/ YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
C +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
C +8X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
C +1X,'S',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
C - H18 -


```plaintext
ClibPDF - www.fastio.com

Jan 25 15:17 1985  qi21 Page 2

+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, QBYTE/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
C ==1================2================3================4================5================6================7==
C
C -----------------------------------REVISION HISTORY---------------------------------------
C
C LEVEL AUTHOR DATE DESCRIPTION
C ==1==2==3==4==5==6==7==
C
C .01D. SL 85/01/25. REVISED COMMENTS.
C
C
C ==1================2================3================4================5================6================7==
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=2
C +,BPW=60,DIM BUF=(1006+64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
C COMMON /MSUG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-------2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(4,8)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01D/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C FORMAT('1QI22',A4,2A9)
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C CALL WRMSUG1(FTRUE)
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *, 'REPORTS ',BUF(2),', ', EOF ',JEOF
C END
C
C=================================================================================================
C SUBROUTINE WRMSUG1(FTRUE)
C IMPLICIT INTEGER(A-E,G-Z)
C DIMENSION FTRUE(37)
C PRINT 100, (FTRUE(I),I=1,5)
C +,((FTRUE(S-J+1)*4+I),J=1,8),I=1,4)
C 100 FORMAT('/ YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
C +, ' BOX10 ',F4.0,' CHECKSUM ',F6.0/
C +8X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
C +1X,'W',2F8.2,F8.0,F8.2,2F8.0,2F8.1/
C
- H20 -
CLIBPDF - www.fastio.com

Jan 25 15:18 1985  qi22 Page 2

+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/
C
DATA FBASE/1799,4*0
+,2*10221,-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 GTRPT 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
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===
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=250,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=960,ID=0
C +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=58)
C
C COMMON /DSU2/FUNITS(58),FBASE(58),BITS(58),OFFSET(58)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C C-----2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(8,6)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 4,LEVEL,DTE,TME
C FORMAT('1QI24','A4,2A9')
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C PRINT 300,FTRUE
C 300 FORMAT(/' DECADE ',F4.0, ' MONTH ',F3.0, ' BOX2 ',F6.0, ' BOX10 '
C +,F4.0, ' CHECKSUM ',F6.0/
C +,8X,'0',7X,'1',7X,'2',7X,'3',7X,'4',7X,'5',7X,'6',7X,'N'/
C +,1X,'S',7F8.2,F8.0/
C +,1X,'A',7F8.2,F8.0/
C +,1X,'U',7F8.2,F8.0/
C +,1X,'V',7F8.2,F8.0/
C +,1X,'P',7F8.2,F8.0/
C +,1X,'R',7F8.1,F8.0/
C +,1X,'U',F8.2,' V ',F8.2,' UV ',F8.2,' UU ',F8.2,' VV ',F8.2)
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *, ' REPORTS ',BUF(2),',',EOF ',JEOF
C END

-H22-
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
C ==1==1==2==2==3==3==4==4==5==5==6==6==7==
C
C--------REVISION HISTORY-----------------------------
C
LEVEL AUTHOR DATE DESCRIPTION
C
C==1==1==2==2==3==3==4==4==5==5==6==6==7==
C
C .01C. SL 85/01/25. REVISED COMMENTS.

C==1==1==2==2==3==3==4==4==5==5==6==6==7==
C
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=60,RPTOFF=1,FMISS=-9999.9,INDEXCK=5,BPR=3712,ID=0
C +,BPW=60,DIM BUFF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=271)
C
C COMMON /MST3/FUNITS(271),FBASE(271),BITS(271),OFFSET(271)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C----2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(19,14)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C FORMAT('1QL14','A4','2A9')
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C PRINT 300,(FTRUE(I),I=1,5)
C 300 FORMAT(//'YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
C +,' BOX10 ',F4.0,' CHECKSUM ',F6.0/
C +9X,'X','X','X','X','X','X','X','X','X','X','X','X','X','X','X','X'
C +,7X,'0',7X,'1',7X,'2',7X,'3',7X,'4',7X,'5',7X,'6')
C PRINT 301,((FTRUE2(I,J),J=1,14),I=1,19)
C 301 FORMAT(1X,'S',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'A',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'W',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'U',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'V',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'P',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'C',F8.1,3F8.2,F8.0,9F8.1/
C +1X,'Q',F8.1,3F8.2,F8.0,9F8.2/
C +1X,'R',F8.1,3F8.2,F8.0,9F8.1/

- H24 -
+1X, 'S-A        ,F8.1,3F8.2,F8.0,9F8.2/
+1X, '(S-A)*W   ',F8.1,3F8.2,F8.0,9F8.1/
+1X, 'Q*S-Q      ',F8.1,3F8.2,F8.0,9F8.2/
+1X, '(Q*S-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,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.01,7*.1
  +,6*.01,.01,.01,.01,.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

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

- H25 -
- H26 -

SEE Q19 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+=---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 ===1+=---3+===4+---5+---6+---7+===
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=250, RPTOFF=1, FMISS=0., INDEXCK=5, BPR=256, ID=0
C +, BPW=60, DIM BUF=(1006*64-1)/BPW+1, DIM PK=(BPR-1)/BPW+1, DIM UN=23)
C
C COMMON /TRP1/FUNITS(23), FBASE(23), BITS(23), OFFSET(23)
C
C DIMENSION BUF(DIM BUF), PK(DIM PK), UN(DIM UN), FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(6,3)
C EQUIVALENCE (FTRUE(6), FTRUE2)
C
C DATA LEVEL/4H.01C/, BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1, LEVEL, DTE, TME
C FORMAT(1QL16', A4, 2A9)
C
C 100 CALL GETRPT(1, FMISS, FUNITS, FBASE, BITS, OFFSET, INDEXCK, ID
C +, BPR, BPW, RPTOFF, BUF, DIM BUF, PK, DIM PK, UN, DIM UN, FTRUE, JEOF)
C IF(JEOF.NE.0) GOTO 900
C
C PRINT 300, FTRUE
C 300 FORMAT(1/ YEAR ', F5.0, ’ MONTH ', F3.0, ’ BOX2 ', F6.0, ’ BOX10 ', F4.0
C +, ’ CHECKSUM ', F5.0/
C +, ’1X, ’NI ', ’6F8.0/
C +, ’1X, ’NL ', ’6F8.0/
C +, ’1X, ’NU ', ’6F8.0/
C IF(BUF(2) .LT. MAX) GOTO 100
C
C 900 PRINT *, ’, REPORTS ’, BUF(2), ’, EOF ’, JEOF
C END
C
C=================================================================
C BLOCK DATA TRP1
C IMPLICIT INTEGER(A-E,G-Z)
C
- H27 -
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 Q19 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----

- H28 -
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 ==1==2==3==4==5==6==7==
C .01C. SL 85/01/25. REVISED COMMENTS.
C
C ==1==2==3==4==5==6==7==
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=300,RPTOFF=1,FMISS=-999.9,INDEXCK=35,BPR=192,ID=0
C +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=35)
C
C COMMON /CMR5/FIELD(35),FTRUEL(35),FTRUEU(35),FUNITS(35)
C +,FBASE(35),BITS(35),OFFSET(35)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C 1 FORMAT('QL21',A4,2A9)
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C PRINT 300,(FIELD(I),FTRUE(I),I=1,DIM UN)
C 300 FORMAT(6(IX,A5,F7.1))
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *,', REPORTS ',',BUF(2)',', EOF ',',JEOF
C END
C
C ================================================================
C BLOCK DATA CMR5
C IMPLICIT INTEGER(A-E,G-Z)
C
C COMMON /CMR5/FIELD(35),FTRUEL(35),FTRUEU(35),FUNITS(35)
C +,FBASE(35),BITS(35),OFFSET(35)
C
C DATA FIELD/8HBOX10 ,8HMUNTH ,8HBOX2 ,8HYEAR ,8HDAY ,
C +8HHOUR ,8HX ,8HY ,8HS ,8HBI ,8HA ,
C +8HDP ,8HTI ,8HU ,8HV ,8HDI ,8HWI ,
C +8HP ,8HC ,8HNH ,8HCL ,8HH ,8HHI ,
C +8HCM ,8HCH ,8HST ,8HPW ,8HCD ,8HLF ,
C - H29 -
C
DATA FTRUE/3*1.,1800.,1.,3*0.,-5.,0.,-88.,2*0.,2*-102.2,2*0.,870.
+.17*0. /
C
+.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
C
C
RPTOFF 1
DATA OFFSET/
+ 0, 10, 14, 28, 36, 41, 46, 51, 56, 65, 67, 78, 88, 91,102,113
+.116,118,129,133,137,141,145,147,151,155,159,166,176,179,181
+.183,185,187 /
C

=========================================

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

- H30 -
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
C==1=1==2====3==4======5==6======7==

C-----------------------REVISION HISTORY---------------------------------
C
C
C==1==2===3======4====5=====6======7==
LEVEL AUTHOR DATE DESCRIPTION
C
C==1==2===3======4====5=====6======7==

C
C==1==2===3======4====5=====6======7==
IMPLICIT INTEGER(A-E,G-Z)
C
C
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
C
COMMON /MSTG1,FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
C
DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C------2 DIMENSIONAL FTRUE
DIMENSION FTRUE2(4,8)
EQUIVALENCES (FTRUE(6),FTRUE2)
C
C
DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
C
CALL DATE(DTE)
CALL TIME(TME)
PRINT 1,LEVEL,DTE,TME
1 FORMAT(’1QL28’,A4,2A9)
C
C
100 CALL GETRPT1,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
C
CALL WRMSTG1(FTRUE)
IF(BUF(2).LT.MAX)GOTO 100
C
C
900 PRINT *, ’REPORTS ’,BUF(2), ’, EOF ’,JEOF
END
C
C=============================================
SUBROUTINE WRMSTG1(FTRUE)
IMPLICIT INTEGER(A-E,G-Z)
DIMENSION FTRUE2(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/

-H31-
C...............GROUP 3

BLOCK DATA MSTG1
IMPLICIT INTEGER(A-E,G-Z)

COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)

DATA FUNITS/1., 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.E-2, 0.1, 1.E-2, 0.1
+0.1, 0.1, 0.1
+0.2, 0.2, 0.2
+0.2, 0.2, 0.2

DATA FBASE/1799., 0., 0., 0., 0.
-1., 0., 0.
-1., -1.
-1.
-5., -5.
-5.

DATA BITS/8,4,14,10,12,16*16,16*4/

DATA OFFSET
+16, 24, 28, 42, 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 -----

-H32-
PROGRAM QL29
C-----READ AND PRINT MSTG1 GROUP 4
C
C-----RPTIN, BUFFER IN, UNIT, LENGTH, CBYTE/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 =1==2==3==4==5==6==7==
C .01C. SL 85/01/25. REVISED COMMENTS.

C
C ==1==2==3==4==5==6==7==
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,Id=4
C +,BPW=60,DIM BUF=(1006+64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
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
C DIMENSION FTRUE2(4,8)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
CALL DATE(DTE)
CALL TIME(TME)
PRINT 1,LEVEL,DTE,TME
C
FORMAT(’1QL29’,A4,2A9)
C
100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JE0F)
IF(JEOF.NE.0)GOTO 900
C
CALL WRMSTG1(FTRUE)
IF(BUF(2).LT.MAX)GOTO 100
C
900 PRINT ’,’ REPORTS ’,BUF(2),’,’, EOF ’,JE0F
END
C==============================================
SUBROUTINE WRMSTG1(FTRUE)
IMPLICIT INTEGER(A-E,G-Z)
DIMENSION FTRUE2(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/

-H33-
C

C========================================================================

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.
+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/

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 Q19 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
C==1==2==3==4==5==6==7===
C
C-------REVISION HISTORY------------------------------------------
C
C
C
C==1==2==3==4==5==6==7===
C
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=5
C +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
C COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(4,8)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C FORMAT(1QL30',A4,2A9)
C
C 100 CALL GETRPT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C CALL WRMSTG1(FTRUE)
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *,' REPORTS ',BUF(2),',',EOF ',JEOF
C END
C
C=====================================================================
C SUBROUTINE WRMSTG1(FTRUE)
C IMPLICIT INTEGER(A-E,G-Z)
C DIMENSION FTRUE(37)
C PRINT 100,(FTRUE(I),I=1,5)
C +(,(FTRUE(5+(J-1)*4+I),J=1,8),I=1,4)
C 100 FORMAT(1, YEAR ',F5.0, MONTH ',F3.0, BOX2 ',F6.0
C +, BOX10 ',F4.0, CHECKSUM ',F6.0/
C +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
C +1X,'C',2F8.1,F8.0,F8.1,F8.0,3F8.1/
C BLOCK DATA MSTG1
  IMPLICIT INTEGER(A-E,G-Z)
  COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
  DATA FUNITS/1., 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/
  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/
  DATA BITS/8, 4, 14, 10, 12, 16*16, 16*4/
  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

----- 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
C==1==2==3==4==5==6==7==
C
C---------------------------------------------------
C LEVEL AUTHOR DATE DESCRIPTION
C ==01==0C==85/01/25. REVISED COMMENTS.
C
C---------------------------------------------------
C
C==1==2==3==4==5==6==7==
C IMPLICIT INTEGER(A-E,G-Z)
C
C PARAMETER(MAX=400,RPTOFF=1,FMISS=-9999.,INDEXCK=5,BPR=384,ID=6
C +,BPW=60,DIM BUF=(1006*64-1)/BPW+1,DIM PK=(BPR-1)/BPW+1,DIM UN=37)
C
C COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)
C
C DIMENSION BUF(DIM BUF),PK(DIM PK),UN(DIM UN),FTRUE(DIM UN)
C
C-----2 DIMENSIONAL FTRUE
C DIMENSION FTRUE2(4,8)
C EQUIVALENCE (FTRUE(6),FTRUE2)
C
C DATA LEVEL/4H.01C/,BUF/DIM BUF*0/
C
C CALL DATE(DTE)
C CALL TIME(TME)
C PRINT 1,LEVEL,DTE,TME
C 1 FORMAT('1QL31','A4,2A9)
C 100 CALL GETRT(1,FMISS,FUNITS,FBASE,BITS,OFFSET,INDEXCK,ID
C +,BPR,BPW,RPTOFF,BUF,DIM BUF,PK,DIM PK,UN,DIM UN,FTRUE,JEOF)
C IF(JEOF.NE.0)GOTO 900
C
C CALL WRMSTG1(FTRUE)
C IF(BUF(2).LT.MAX)GOTO 100
C
C 900 PRINT *,R REPORTS *,BUF(2),EOF*,JEOF
C END
C
C======================================================================
C SUBROUTINE WRMSTG1(FTRUE)
C IMPLICIT INTEGER(A-E,G-Z)
C DIMENSION FTRUE(37)
C PRINT 100,(FTRUE(I),I=1,5)
C *((FTRUE(J+1)*4+I),J=1,8),I=1,4)
C 100 FORMAT(/' YEAR ',F5.0,' MONTH ',F3.0,' BOX2 ',F6.0
C +' BOX10 ',F4.0,' CHECKSUM ',F6.0/
C +9X,7X,'3',7X,'M',7X,'N',7X,'E',7X,'D',7X,'H',7X,'X',7X,'Y'/
C +9X,'S-A ',2F8.2,F8.0,F8.2,F8.0,3F8.1/
C -H37-
END

C=================================GROUP 6=================================

BLOCK DATA MSTG1
IMPLICIT INTEGER(A-E,G-Z)

COMMON /MSTG1,FUNITS(37),FBASE(37),BITS(37),OFFSET(37)

DATA FUNITS/1., 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.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/

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/

DATA BITS/8,4,14,10,12,16*16,16*4/

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=================================GROUP 6=================================

----- 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
C       =1==2==3==4==5==6==7==
C
C------- REVISION HISTORY -------------------------------
C       =1==2==3==4==5==6==7==
C       LEVEL AUTHOR DATE DESCRIPTION
C
C .01C. SL 85/01/25. REVISI...
C ============== GROUP 7 ==============

BLOCK DATA MSTG1
IMPLICIT INTEGER(A-E,G-Z)

COMMON /MSTG1/FUNITS(37),FBASE(37),BITS(37),OFFSET(37)

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/

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.
+0.5, 0.5, 0.5, 0.5
+-0.5, -0.5, -0.5, -0.5/

DATA BITS/8,4,14,10,12,16*16,16*4/

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 Q19 FOR LISTINGS OF SUBROUTINES GETRPT AND RCDIN -----
CONVERTED BY CONVRT: TSCon.01B
PROGRAM RDIVN

Purpose - READ PACKED INVENTORIES FOR PRE-70'S OR 70'S DATA MADE BY PROGRAM DUPELIM
Written by - JANE HISCOX

LEVEL AUTHOR DATE DESCRIPTION
00210
.01B. SL 85/01/30. REVISED COMMENTS; CONVERT FROM TIMESHARING FORTRAN.
00240
00250
00260
00270
00280
00290
00300
00310
00320
00330
00340
00350
00360
00370
00380
00390
00400
00410
00420
00430
00440
00450
00460
00470
00480
00490
00500
00510
00520
00530
00540
00550
00560
00570
00580
00590
00600
00610
00620
00630
00640
00650

IMPLICIT INTEGER (A-Z)
CHARACTER*4 LEVEL

DIMENSION STORE (5000), CARD (50)

COMMON /QC/ INVNF (14, 11)

DATA LEVEL /'01B'/, NSTORE, NSID, NCD, NDS/5000, 24, 50, 8/
DATA RQC, CQC/14, 11/, BITBOX, BITYR, BITIOD, BITGT /10, 8, 15,
+20/
DATA IU, JU, OU /1, 2, 5/
DATA CARD /110, 116, 117, 118, 119, 128, 143, 150, 151, 152, 155,
+156, 184, 185, 186, 187, 188, 189, 192, 193, 194, 195,
+196, 197, 281, 555, 666, 849, 850, 876, 877, 878, 879,
+880, 881, 882, 888, 889, 891, 897, 898, 899, 900, 901,
+902, 926, 927, 928, 999, 50/

REWIND IU
REWIND JU
REWIND OU

DTE = DATE (K)
TME = TIME (K)
READ (JU,*, END=900) BOX
WRITE (5,5) BOX, LEVEL, DTE, TME
5 FORMAT ('I INVENTORIES FOR BOX ',I3,60,'BY RDINV',A,2X,2A10)

100 BUFFER IN (IU,0) (STORE(1), STORE(NSTORE))
IF (UNIT(IU) .LT. 0) THEN
  OFF = 0
  NWORD = 1
  CALL GBYTE (STORE(NWORD), BOX10, OFF, BITBOX)
IF (BOX10 .EQ. BOX) THEN
  OFF = OFFSET (OFF, NWORD, BITBOX)
175 CALL GBYTE (STORE(NWORD), YEAR, OFF, BITYR)
OFF = OFFSET (OFF, NWORD, BITYR)
IF (YEAR .NE. 0) THEN
  YEAR = YEAR + 1799

-H41-
WRITE (5,200) YEAR
FORMAT ("//' YEAR = ',I4,'X, 'MO. IN OUT UNCERTAIN', /I1X,26(=')")
SUMI = 0
SUMO = 0
SUMD = 0
DO 225 MO = 1,12
   CALL GETNUM (STORE, IMO, OFF, NWORD, BITIOD)
   CALL GETNUM (STORE, OMO, OFF, NWORD, BITIOD)
   CALL GETNUM (STORE, DMO, OFF, NWORD, BITIOD)
   IF (IMO .NE. 0) WRITE (5,210) MO, IMO, OMO, DMO
210 FORMAT (1X,I2,1X,2I6,3X,I6)
   SUMI = SUMI + IMO
   SUMO = SUMO + OMO
   SUMD = SUMD + DMO
225 CONTINUE
WRITE (5,250) SUMI, SUMO, SUMD
250 FORMAT (1X,26(=')/4X,2I6,3X,I6)
C
--------UNPACK YEARLY TOTALS FOR SOURCE IDS
WRITE (5,260)
260 FORMAT ("//' TOTALS BY SID',/
   + 1X,'SID IN OUT UNCERTAIN',/1X,
   + 36(=')")
SUMI = 0
SUMO = 0
SUMD = 0
DO 300 JR = 1,NSID
   CALL GETNUM (STORE, ISID, OFF, NWORD, BITIOD)
   CALL GETNUM (STORE, OSD, OFF, NWORD, BITIOD)
   CALL GETNUM (STORE, DSD, OFF, NWORD, BITIOD)
   IF (ISID .NE. 0) WRITE (5,275) JR, ISID, OSD, DSD
275 FORMAT (1X,I3,3(3X,I7))
   SUMI = SUMI + ISID
   SUMO = SUMO + OSD
   SUMD = SUMD + DSD
300 CONTINUE
WRITE (5,325) SUMI, SUMO, SUMD
325 FORMAT (1X,36(=')/4X,3(3X,I7))
GO TO 175
ENDIF
C
--------UNPACK GRAND TOTALS BY SID
WRITE (5,350) BOX10
350 FORMAT ("'1 GRAND TOTALS FOR BOX ',I3,/
   + 1X,'SID IN OUT UNCERTAIN',/1X,
   + 36(=')")
SUMI = 0
SUMO = 0
SUMD = 0
DO 400 JR = 1,NSID
   CALL GETNUM (STORE, ISID, OFF, NWORD, BITGT)
   CALL GETNUM (STORE, OSD, OFF, NWORD, BITGT)
   CALL GETNUM (STORE, DSD, OFF, NWORD, BITGT)
   IF (ISID .NE. 0) WRITE (5,275) JR, ISID, OSD, DSD
   SUMI = SUMI + ISID
400 CONTINUE
SUMO = SUMO + OSID
SUMD = SUMD + DSID

400 CONTINUE
WRITE (5,325) SUMI, SUMO, SUMD

C
C -------------------- UNPACK GRAND TOTALS BY CARD DECK
WRITE (5,500)
FORMAT ('///,1X,' CD IN OUT UNCERTAIN',/1X,
+ 36('='))
SUMI = 0
SUMO = 0
SUMD = 0
DO 600 JR = 1,NCD
CALL GETNUM (STORE, ICD, OFF, NWORD, BITGT)
CALL GETNUM (STORE, OCD, OFF, NWORD, BITGT)
CALL GETNUM (STORE, DCD, OFF, NWORD, BITGT)
IF (ICD .NE. 0) WRITE (5,275) CARD(JR), ICD, OCD, DCD
SUMI = SUMI + ICD
SUMO = SUMO + OCD
SUMD = SUMD + DCD
600 CONTINUE
WRITE (5,325) SUMI, SUMO, SUMD

C
C -------------------- UNPACK GRAND TOTALS
WRITE (5,625)
FORMAT ('/// GRAND TOTALS')
CALL GETNUM (STORE, IGT, OFF, NWORD, BITGT)
CALL GETNUM (STORE, OGT, OFF, NWORD, BITGT)
CALL GETNUM (STORE, DGT, OFF, NWORD, BITGT)
WRITE (5,650) IGT, OGT, DGT
650 FORMAT ('/' TOTAL IN = ','I7', ' TOTAL OUT = ','I7,
+ ' NUMBER OF UNCERTAIN IN OUT = ','I7)

C
C -------------------- UNPACK TOTALS BY DS
WRITE (5,675)
FORMAT ('/// TOTALS BY DUPLICATE STATUS',/5X,
+ ' DS TOTAL',/5X,12('='))
SUMDS = 0
DO 700 JR = 1,NDS
CALL GETNUM (STORE, DDS, OFF, NWORD, BITGT)
J = JR - 1
WRITE (5,685) J, DDS
685 FORMAT (5X,I3,I7)
SUMDS = SUMDS + DDS
700 CONTINUE
WRITE (5,725) SUMDS
725 FORMAT (5X,12('='),/8X,I7)

C
C -------------------- UNPACK QC INVENTORIES
DO 800 JC = 1,CQC
DO 775 JR = 1,RQC
CALL GETNUM (STORE, INVNF(JR,JC), OFF, NWORD, BITGT)
775 CONTINUE
800 CONTINUE
C
CALL PRINVN (BOX10)
GO TO 900

- H43 -
ENDIF
GO TO 100
ENDIF
900 REWIND IU
REWIND JU
REWIND OU
END

******************************************************************************

SUBROUTINE GETNUM (STORE, NUM, OFF, NWORD, BITS)

--------------- UNPACK NUMBER, UPDATE OFFSET. IF THE UNPACKED NUMBER
IS THE MAXIMUM SIZE FOR NUMBER OF BITS, UNPACK THE NEXT
NUMBER AND SUM THEM.

STORE - ARRAY TO UNPACK NUMBER FROM
NUM   - RESULTANT NUMBER
OFF   - OFFSET
NWORD - WORD OF ARRAY STORE TO UNPACK FROM
BITS  - NUMBER OF BITS TO UNPACK FROM STORE

IMPLICIT INTEGER (A-Z)
DIMENSION STORE (*)

NUM = 0
100 CALL GBYTE (STORE(NWORD), N, OFF, BITS)
OFF = OFFSET (OFF, NWORD, BITS)
NUM = NUM + N
IF (N .GE. (2**BITS - 1)) GO TO 100
END

******************************************************************************

INTEGER FUNCTION OFFSET (OFF, NWORD, BITS)

--------------- UPDATE OFFSET AND NWORD BY BITS

IMPLICIT INTEGER (A-Z)
DATA WRDSIZ / 60/

OFFSET = OFF + BITS
IF (OFFSET .GE. WRDSIZ) THEN
   OFFSET = OFFSET - WRDSIZ
   NWORD = NWORD + 1
ENDIF
END

******************************************************************************

SUBROUTINE PRINV (BOX10)

--------------- PRINT QC INVENTORIES

IMPLICIT INTEGER (A-Z)
CHARACTER FLAG (14)*8

- H44 -
C
COMMON /QC/ INVNF (14,11)
C
DATA FLAG /'SHIP POS','WIND ','VIS ','PRES WX ','PAST WX ',
+ 'PRESSURE','DRY BULB','WET BULB','DEW PT ','SEA TEMP',
+ 'CLOUDS ','WAVES ','SWELLS ','P TEND '/
C
WRITE (5,10) BOX10
10 FORMAT (///,' QUALITY CONTROL FLAGS, BOX10 = ',I3,
+ /1X,'FLAG/VALUE',3X,'MISSING',7X,'R',9X,'A',9X,'B',9X,
+ 'J',9X,'K',9X,'L',9X,'M',9X,'N',9X,'Q',9X,'S',5X,
+ 'TOTAL')
   DO 230 JR = 1,14
      TOTAL = 0
   DO 220 JC = 1,11
      TOTAL = TOTAL + INVNF(JR,JC)
   220 CONTINUE
      WRITE (5,225) FLAG(JR),(INVNF(JR,JC),JC=1,11),TOTAL
225 FORMAT (1X,A,12I10)
   230 CONTINUE
   END

-H45-
CONVERTED BY CONVRT: TSCON.01B

SUBROUTINE READER(UNIT,TARGET)

-----------READ LANDLOCKED BOX2 MAP INTO TARGET(16202)
FROM INTEGER &UNIT.

1H. = LAND
1H* = COASTAL
1H = SEA

-----------REVISION HISTORY---------------------

LEVEL AUTHOR DATE DESCRIPTION

.01A. SDW 85/02/15. ORIGINAL VERSION TAKEN FROM LLLIBS.01J.
.01B. SL 85/02/15. REPLACE ALL R1 FORMAT DESCRIPTORS WITH
A1. REMOVE CONVERT TO INTEGER ENTRY.

REMOVE ALL END= FROM READ STATEMENTS.

REVISED COMMENTS. CONVERT FROM
TIMESHARING FORTRAN.

IMPLICIT INTEGER(A-E,G-Z)
DIMENSION TARGET(16202)

-----------READ, TARGET WILL REMAIN IN A1 WITH NO CONVERSION

READ(UNIT,100) TARGET(1)

100 FORMAT(/,,6X,A1)

DO 300 KLAT=1,90
   KLO1=(KLAT-1)*180+2
   KLO2=KLO1+89
   READ(UNIT,200) (TARGET(I),I=KLO1,KLO2)

200 FORMAT(6X,90A1)

300 CONTINUE

READ(UNIT,350)

350 FORMAT(3(/))

DO 500 KLAT=1,90
   KLO1=(KLAT-1)*180+92
   KLO2=KLO1+89
   READ(UNIT,200) (TARGET(I),I=KLO1,KLO2)

500 CONTINUE

READ(UNIT,600) TARGET(16202)

600 FORMAT(95X,A1)
END