National Aeronautics and Space Administration
Goddard Earth Science
Data Information and Services Center (GES DISC)

README Document for the Nimbus-7 Stratospheric and Mesospheric Sounder (SAMS) Level 1 Radiance Data

SAMSN7L1RAT

Last Revised 08/14/2019

Goddard Earth Sciences Data and Information Services Center (GES DISC)
https://disc.gsfc.nasa.gov
NASA Goddard Space Flight Center
Code 610.2
Greenbelt, MD 20771 USA

Prepared By:

James E. Johnson	08/14/2019
Name GES DISC GSFC Code 610.2	Date

Reviewed By:

Name	mm/dd/yyyy
Name GSFC Code xxx	Date

Name	mm/dd/yyyy
Name GSFC Code xxx	Date

Goddard Space Flight Center
Greenbelt, Maryland

Revision History

Revision Date	*Changes*	*Author*
08/14/2019	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Nimbus-7 Stratospheric and Mesospheric Sounder (SAMS) Level-1 Radiance Data product.

1.1 Data Product Description

The Nimbus-7 Stratospheric and Mesospheric Sounder (SAMS) Level-1 Radiance Data product contains uncalibrated and calibrated data frame by frame along the orbit, housekeeping functions, derived tangent point pressures, relevant spacecraft functions, and retrieved temperature profiles at 2.5-deg intervals along the tangent track. The instrument observed emission from the limb of the atmosphere through seven pressure-modulated cells and six detectors in order to determine temperature and vertical concentrations of H_2O, N_2O, CH_4, CO, and NO in the stratosphere and mesosphere.

Each file contains roughly one days worth of data (~14 orbits per day). SAMS is a limb profiler and spatial coverage is near global between latitude -51 and +70 degrees. Vertical coverage is from about 15 to 140 km (O3 channel to 65 km), with vertical resolution of about 100 km by 10 km at the limb. The data are available from 24 October 1978 through 13 June 1983. The principal investigators for the SAMS experiment were Prof. John T. Houghton and Dr. Fredric W. Taylor from Oxford University.

This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name SAMS Radiance Archive Data (SAMS/RAT) with the identifier ESAC-00243 (old id 78-098A-02A).

1.1.1 The Stratospheric and Mesospheric Sounder

The objective of the Nimbus-7 Stratospheric and Mesospheric Sounder (SAMS) experiment was to observe emission from the limb of the atmosphere through seven pressure-modulated cells and six detectors in order to determine temperature and vertical concentrations of H2O, N2O, CH4, CO, and NO in the stratosphere and mesosphere. Measurements of zonal wind in this region were attempted by observing the Doppler shift of atmospheric emission lines. Radiation from the limb of the atmosphere was incident on a scan mirror in front of a 15-cm aperture telescope. The scan mirror scanned the limb, viewed space for calibration, and viewed the atmosphere obliquely to obtain vertical profiles at three adjacent fields of view (100 km by 10 km at the limb). The instrument operated successfully from October 1978 until June 1983. The experiment was based on techniques used on the previously flown Selective Chopper Radiometers (SCR) on Nimbus 4 and 5, and the Pressure Modulated Radiometer (PMR) flown on Nimbus 6. The Improved Stratospheric and Mesospheric Sounder (ISAMS) was flown on the Upper Atmosphere Research Satellite in 1991.

1.1.2 Nimbus-7 Overview

The Nimbus-7 satellite was successfully launched on October 24, 1978 and was the final in the Nimbus series. The spacecraft included nine experiments: (1) the Limb Infrared Monitor of the Stratosphere (LIMS) for making vertical profiles of temperature and concentrations of O_3, H_2O, NO_2, and HNO_3, (2) a Stratospheric and Mesospheric Sounder (SAMS) providing vertical concentrations of H_2O, CH_4, CO and NO and measure the temperature in the upper atmosphere, (3) the Coastal-Zone Color Scanner (CZCS) for mapping ocean chlorophyll concentrations, (4) the Stratospheric Aerosol Measurement II (SAM II) to map the concentration and optical properties of aerosols, (5) the Earth Radiation Budget (ERB) for measuring the incoming and outgoing reflected and emitted radiation of the Earth, (6) a Scanning Multichannel Microwave Radiometer (SMMR) to obtain and use ocean momentum and energy-transfer parameters on a nearly all-weather operational basis., (7) a Solar Backscatter UV (SBUV) spectrometer to determine the vertical distribution of ozone, (8) the Total Ozone Mapping Spectrometer (TOMS) for mapping the total column amount of ozone, and (9) the Temperature Humidity Infrared Radiometer (THIR) for measuring daytime and nighttime surface and cloudtop temperatures, as well as the water vapor content of the upper atmosphere.

The orbit of the satellite can be characterized by the following:

- circular orbit at ~950 km
- inclination of 99 degrees
- period of an orbit is about 104 minutes
- orbits cross the equator at 26 degrees of longitude separation
- sun-synchronous

1.2 Algorithm Background

The Nimbus-7 SAMS data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers using a 24-bit words, and copied to 1600 bpi 9-track tapes for archival. Further information on the SAMS instrument and data processing can be found in the Nimbus-7 Users' Guide Section 6.

1.3 Data Disclaimer

The data should be used with care and one should first read the Nimbus-7 User’s Guide, Section 6 describing the SAMS experiment. Users should cite this data product in their research.

2. Data Organization

The Nimbus-7 Stratospheric and Mesospheric Sounder Level-1 Radiance Data spans the time period from October 26, 1978 to June 9, 1983. Each file typically contains one orbit of data.

2.1 File Naming Convention

The data product files are named according to the following convention:

<Platform>-<Instrument>_<Level>-<Product>_<DateTime>_<TapeNumber>.<Suffix>

where:
o Platform = name of the platform or satellite (Nimbus7)
o) Instrument = name of the instrument and product (SAMS)
o) Level = process level (L1)
o) Product = radiance archive tape (RAT)
o) Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmm> where

1. YYYY = 4 digit year (1978 – 1979)
2. MM = 2 digit month (01-12)
3. DD = 2 digit day of month (01-31)
4. hh = 2 digit hour (00-23)
5. mm = 2 digin minute (00-23)

o TapeNumber = 5 digit number of tape (preceded by ‘DD’ - primary or ‘DC’ - backup)
o Suffix = the file format (always TAP, indicating tape binary data)

```
File name example: Nimbus7-SAMS_L1-RAT_1978m1026t2040_DD58666.TAP
```

2.2 File Format and Structure

The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. The files were written on the original 1600 bpi 9-track tapes using a blocked FORTRAN format. The first file on the tape is the tape header file with two records containing text encoded information about the tape. This is followed by a set of daily data files. Each daily file on the tape contains a set of records with a FORTRAN record size word, the record block, and a FORTRAN record trailing size word. There are about 400 record blocks in a daily file. Record blocks are 4104 bytes, containing a header (260 bytes), and a set of data measurement records (388 bytes each). The last record block in a daily file is 36 bytes in length. At the end of the file there is an End-of-File word (the last file on the tape will end with a double End-of-File word). The data measurement records will contain the radiances, and some may contain derived temperature values. All data use 16-bit words. For the contents and layout of the documentation, see section 3.1.

During data recovery a total of 1253 daily data files were retrieved from the tapes, with 1167 files from the primary DD tapes, and just 86 files from the backup DC tapes. Most of the files (76) from

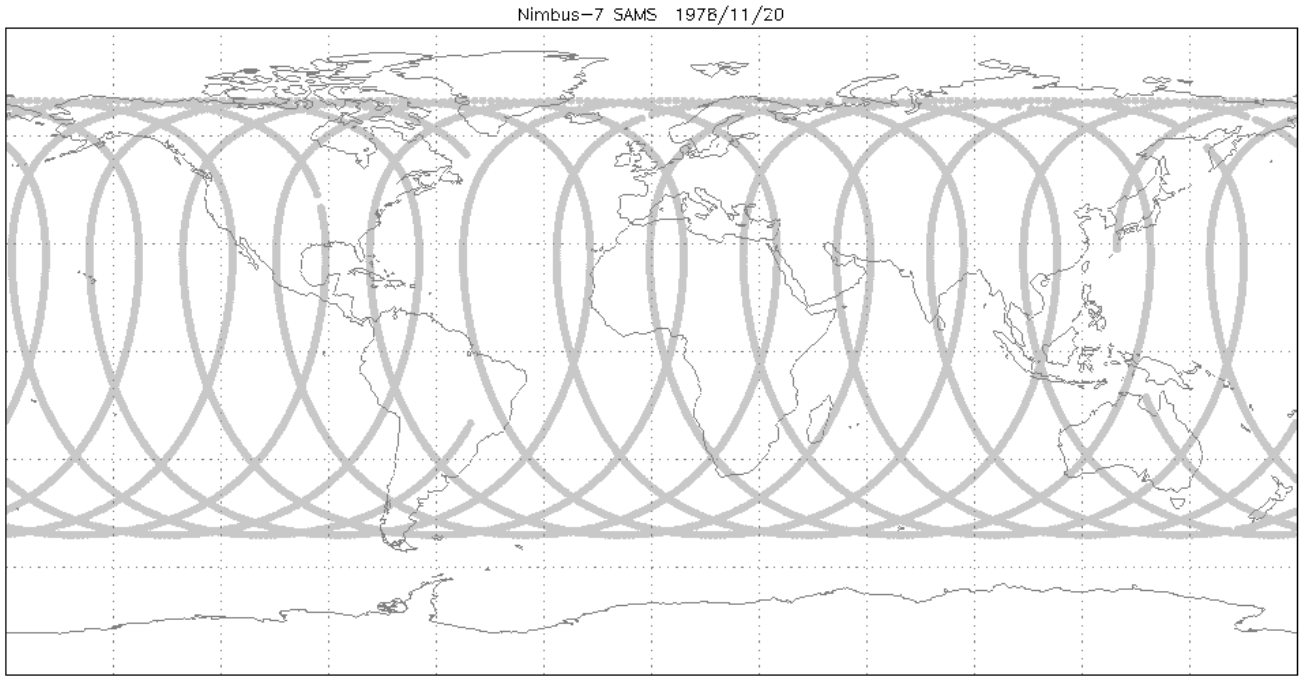
the DC tapes were identical copies of files from the DD tapes, two contained missing records which were complete on the DD tape file counterparts, two contained extra records missing on the DD tape file counterparts, and only 8 _DC files were kept, and 2 _DD files were deleted. In summary, a total of 1175 SAMS daily data files were retrieved as part of this data collection and are archived at the GES DISC.

It should be noted that these data are also available in a more recent and complete collection in the Oxford University created SAMS/Nimbus-7 Level 1 Radiance Data from CDROM data product also available at the GES DISC. This CDROM product also contains temperature and trace gas retrievals.

2.3 Key Science Data Fields

The primary science data fields in this data product are the SAMS radiances from the six pressure modulated cells and six wide band detector channels.

Figure 1: Typical data coverage for a Nimbus-7 SAMS Level 1 RAT daily data file.



3. Data Contents

The granularity of this data product is one day.

3.1 Data Records

The Nimbus-7 User's Guide does not describe the layout of the file format. Refer instead to the Tape Specification, number T564011, Rev. C, titled "SAMS Radiance Archival Tape".

The original tape files each included a tape header file. These were then followed by a set of daily data files, which would include typically over 400 record blocks containing the radiance values. As part of the recovery, the GES DISC has extracted and archived the daily data files from the tape. The original data were written on IBM machines using 16-bit words. Each record block consists of a 3 word block header and several data headers, data records and temperature records (see Table 3-1 to 3-4 below). Record blocks are 4104 bytes, the last record block is 36 bytes, followed by a 0 byte EOF word.

Table 3-1 Record Block Header (3 x 16-bit integers)

Word	Field Name	Units	Size	Comments
1	Block Number		1	
2	Block Identifier		1	3: data record block 132: end record block
3	File Number on Tape		1	

Table 3-2: Data Header (260 x 16-bit integers), Record ID = 7201

Word	Field Name	Units	Size	Comments
1	Record Length (in bytes)		1	
2	Record Number		1	
3	Record Identifier		1	7201
4 – 7	System Flags		4	
8	Header Number		1	0 = no data
9 – 13	Reserved		5	
14	Orbit Number (as received)		1	

15	Segment Number (as received)		1	
16	True Orbit Number		1	
17 – 20	Start Time of Data (year, day, seconds)		4	Seconds 2 byte word
21 – 24	End Time of Data (year, day, seconds)		4	Seconds 2 byte word
25 – 28	Production Time of Data (year, day, seconds)		4	Seconds 2 byte word
29 – 32	Last Ascending Eq. Crossing (day, sec, long. x 100)		4	Seconds 2 byte word
33 – 36	Last Descending Eq. Crossing (day, sec, long. x 100)		4	Seconds 2 byte word
37 – 40	Last Day to Night Crossing (day, sec, lat. x 100)		4	Seconds 2 byte word
41 – 44	Last Night to Day Crossing (day, sec, lat. x 100)		4	Seconds 2 byte word
45	Number of Major Frames		1	
46	No. of Checksum Errors on Transmission		1	
47	No. of Checksum Errors on Mag. Tape		1	
48	No. of Sync Errors		1	
49	No. of Verification Readouts		1	
50	No. of length of CDA file (block with calib. info)		1	
51, 52	Mode at Start of Day		2	2 byte word
53, 54	Mode at End of Day		2	2 byte word
55	Lat/Long. Error Flag		1	
56	Nominal of Eigen Coefficients Used in Retrieval		1	
57	Nominal of Temperature Sub-Levels Used		1	
58	Nominal Year of this Sub Block of Data		1	
59	Nominal Day of this Sub Block of Data		1	
60 – 66	PMC Mean Temperature (x 100)	K	7	
67 – 73	PMC Mean Pressure (x 100)	mb	7	
74 – 80	PMC Mean Period	counts	7	
81	Flag to Indicate Data Have Limited Processing		1	
82 – 212	Spares		131	all zeroes
213	Version Number of Program (x 10)		1	

214	Version Number of Data Format		1	
215 – 259	Spares		45	all zeroes
260	Checksum		1	

Table 3-3: Radiance Data (388 x 16-bit integers), Record ID = 7202

Word	Field Name	Units	Size	Comments
1	Record Length (in bytes)		1	
2	Record Number		1	
3	Record Identifier		1	7202
4	Format (data type) and Mark (format gen #)		1	2 bytes
5	Error Flags		1	(see note 1)
6 – 9	Time (year, day, seconds)		4	Seconds 2 byte word
10	Satellite Latitude (x 100)	deg	1	
11	Satellite Longitude (x 100)	deg	1	
12	Satellite Altitude	km	1	
13	Latitude of Tangent Point (x 100)	deg	1	
14	Longitude of Tangent Point (x 100)	deg	1	
15 – 17	Flags		3	(see note 2)
18	Blackbody Temperature (x 100)	deg C	1	
19	Chopper Temperature (x 100)	deg C	1	
20 – 25	Digital B		6	(see note 3)
26 – 30	Spacecraft Status		5	
31 – 48	Channel Identification		18	4 bytes x 9 channels (see note 4)
49 – 144	(Calibrated) Radiances (B1 chans uncalibrated)		96	8 words x 12 slots (see note 5)
145 – 156	Estimated Errors in Radiances		12	
157 – 163	PMC Pressures	mb	7	(see note 6)
164 – 170	PMC Temperatures (x 100)	deg C	7	(see note 6)
171 – 178	Azimuth Position		8	

179 – 186	LVDT (x 1000)	deg	8	
187 – 194	Roll Angle (Chan. D at Step 40)		8	
195 – 202	Tangent Height of Channel D (x 100)	km	8	
203 – 210	Tangent Scale Height of A Channels (x 1000)		8	(see note 7)
211 – 218	Tangent Scale Height of B Channels (x 1000)		8	(see note 7)
219 – 226	Tangent Scale Height of C Channels (x 1000)		8	(see note 7)
227 – 234	Estimated Error on ZT's (x 1000)		8	(see note 7)
235 – 242	Roll Angle Derived by Retrieval (x 10000)	deg	8	
243 – 250	Estimated Errors in Above (x 10000)	deg	8	
251 – 258	ACS Roll Fine Error		8	
259	Terminator Height at Tangent Point (x 100)	km	1	
260 – 267	Yaw		8	16 bytes
268 – 275	Pitch		8	16 bytes
276 – 307	(Uncalibrated) Radiances (A1P/W,A234P/W)		32	4 channels x 8
308 – 355	(Uncalibrated) Radiances (B2P/W,C1P/W,C23P/W)		48	6 channels x 8
356 – 362	PMC Periods		7	
363 – 369	PMC Amplitudes		7	
370	Chopper Amplitude		1	
371 – 377	Sieve Temperatures		7	
378	Chopper Temperature Expanded		1	
379 – 382	A234,B1,A1,C1 Detector Temperatures		4	
383, 384	B1,A324 Detector DC Level		2	
385, 386	M1, M2 Mirror Temperature		2	
387	SEU Temperature A		1	
388	Checksum		1	

Notes 1 to 7: consult the SAMS RAT Description T884021 document.

Table 3-4: Temperature Data (388 x 16-bit integers), Record ID = 7203

Word	Field Name	Units	Size	Comments
1	Record Length (in bytes)		1	
2	Record Number		1	
3	Record Identifier		1	7203
4	Latitude Reference Number		1	Index 1 to 48
5, 6	Time	sec	4	2 byte word
7	Day		1	
8	Year		1	
9	Satellite Latitude (x100)	deg	1	
10	Satellite Longitude (x100)	deg	1	
11	Latitude of Tangent Point (x100)	deg	1	
12	Longitude of Tangent Point (x100)	deg	1	
13	Major Frame in Cal. File corresp. to this Frame		1	
14 –	Eigen Function Coefficients (x10000)		NOE	
14+NOE –	Std. Dev. In Above (x10000)		NOE	
14+2*NOE –	Temperature at Each Level (x10)		NR	
14 + 2*NOE + NR –	Relative Error at Each Level (x10)		NR	
14 + 2*NOE + 2*NR –	Temperature Std. Dev. (x10)		NR	
14 + 2*NOE + 3*NR –	Absolute Temperature at Intervals of 0.2 Log(P)		72	P = 1.4 and up
131 – 387	(repeat words 4 – 130 above 2 times)		2 x 127	
388	Checksum		1	

NOE (Number of Eigen Functions) = 8
NR (Number of Rows) = 10

3.2 Metadata

The metadata are contained in a separate XML formatted file having the same name as the data file with .xml appended to it.

Table 3-2: Metadata attributes associated with the data file.

Name	Description
LongName	Long name of the data product.
ShortName	Short name of the data product.
VersionID	Product or collection version.
GranuleID	Granule identifier, i.e. the name of the file.
Format	File format of the data file.
CheckSumType	Type of checksum used.
CheckSumValue	The value of the calculated checksum.
SizeBytesDataGranule	Size of the file or granule in bytes.
InsertDateTime	Date and time when the granule was inserted into the archive. The format for date is YYYY-MM-DD and time is hh-mm-ss.
ProductionDateTime	Date and time the file was produced in format YYYY-MM-DDThh:mm:ss.ssssssZ
RangeBeginningDate	Begin date when the data was collected in YYYY-MM-DD format.
RangeBeginningTime	Begin time of the date when the data was collected in hh-mm-ss format.
RangeEndingDate	End date when the data was collected in YYYY-MM-DD format.
RangeEndingTime	End time of the date when the data was collected in hh-mm-ss format.
PlatformShortName	Short name or acronym of the platform or satellite
InstrumentShortName	Short name or acronym of the instrument
SensorShortName	Short name or acronym of the sensor
WestBounding Coordinate	The westernmost longitude of the bounding rectangle(-180.0 to +180.0)
NorthBounding Coordinate	The northernmost latitude of the bounding rectangle(-90.0 to +90.0)
EastBounding Coordinate	The easternmost longitude of the bounding rectangle(-180.0 to +180.0)
SouthBounding Coordinate	The southernmost latitude of the bounding rectangle(-90.0 to +90.0)
Orbit	Satellite orbit number.
ElapsedMinTime	Duration in minutes of data collected during an orbit.

4. Reading the Data

The data are written in a binary record-oriented format. Using the record format specification in the section above, users can write software to read the data files. Please note that the data were originally written using a big-endian format, therefore users on little-endian machines will need to swap bytes for the words.

A sample FORTRAN program is included in the Appendix section which will read in the data records. Additionally a FORTRAN function is included to perform byte swapping.

5. Data Services

5.1 GES DISC Search

The GES DISC provides a keyword, spatial, temporal and advanced (event) searches through its unified search and download interface:

https://disc.gsfc.nasa.gov/

5.2 Documentation

The data product landing pages provide information about these data products, as well as links to download the data files and relevant documentation:

https://disc.gsfc.nasa.gov/datacollection/SAMSN7L1RAT_001.html

5.3 Direct Download

These data products are available for users to download directly using HTTPS:

https://acdisc.gesdisc.eosdis.nasa.gov/data/Nimbus7_SAMS_Level1/SAMSN7L1RAT.001/

6. More Information

6.1 Contact Information

Name: GES DISC Help Desk
URL: https://disc.gsfc.nasa.gov/
E-mail: gsfc-help-disc@lists.nasa.gov
Phone: 301-614-5224
Fax: 301-614-5228
Address: Goddard Earth Sciences Data and Information Services Center
Attn: Help Desk
Code 610.2
NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA

6.2 References

"The Nimbus-7 User's Guide - Section 6: The Stratospheric and Mesospheric Sounder (SAMS) Experiment", NASA Goddard Space Flight Center, November 1972, Pages 71-103

Nimbus G, Nimbus Observation Processing System (NOPS) Requirements Document # NG-34, Tape Specification T564011, Rev. C, "SAMS Radiance Archival Tape (RAT)", NASA Goddard Space Flight Center, Greenbelt, MD, 1981

7. Appendices

Acknowledgments

The Nimbus data recovery task at the GES DISC is funded by NASA's Earth Science Data and Information System program.

Acronyms

BCD: Binary Coded Decimal

EOS: Earth Observing System

ESDIS: Earth Science and Data Information System

GES DISC: Goddard Earth Sciences Data and Information Services Center

GSFC: Goddard Space Flight Center

L1: Level-1 Data

SAMS: Stratospheric and Mesospheric Sounder

NASA: National Aeronautics and Space Administration

QA: Quality Assessment

RAT: Radiance Archival Tape

UT: Universal Time

FORTRAN Code

```
C-----------------------------------------------------------------------
C ^NAME: READ_SAMSN7
C   This program will read a Nimbus 7 SAMS Radiance Archival Tape (RAT)
C   Level-1 data file. Each file contains one orbits worth of data.
C
C   The Nimbus 7 SAMS files contain a series of data records. Each of the
C   data records contain the radiances from the 14 SAMS channels, as well as
C   time, geolocation, quality flags and other information. This program will
C   print the contents of each data record.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C     BUFF   - buffer for data record
C     IBLKSZ - size of record block in bytes
C     IOS    - I/O status number
C
C ^NOTES:
C     Compile: gfortran -o READ_SAMSN7.EXE READ_SAMSN7.FOR
C
C ^ORGANIZATION: NASA/GSFC, Code 610.2
C
C ^AUTHOR:  James Johnson
C
C ^ADDRESS: james.johnson@nasa.gov
C
C ^CREATED: June 6, 2019
C-----------------------------------------------------------------------

      CHARACTER*256      FNAME          ! Filename
      CHARACTER          BUFF(4800)     ! Buffer for data record block
      CHARACTER*630      NOPSHD         ! String buffer for NOPS Header (EBCDIC)
      CHARACTER*100      STRBUF         ! String buffer for Tape Header
      INTEGER*4          IBLKSZ         ! Size of records
      INTEGER*4          I4W            ! 4-byte word
      INTEGER*2          I2W            ! 2-byte word
      INTEGER*2          I2WORD(2400)   ! Array of 2-byte words
      CHARACTER          I4TMP(4)       ! Buffer to hold 4-byte word
      CHARACTER          I2TMP(2)       ! Buffer to hold 4-byte word
      EQUIVALENCE        (I4TMP,I4W)
      EQUIVALENCE        (I2TMP,I2W)

C Get the name of the input data file to read
      WRITE (0, *), 'Enter the name of the input file:'
      READ (5,'(A)') FNAME
      PRINT '("File = ",A)', FNAME

C Open the specified input file
      OPEN (UNIT=1, FILE=FNAME, STATUS='OLD', ACCESS='DIRECT',
     &      FORM='UNFORMATTED', RECL=1, ERR=99, IOSTAT=IOS)

C Initialize N (record number) and IOFF (byte offset in file)
      N=0
      IOFF=0
```

```
C Loop through the file reading all records in file
   10 DO
C Read the first 4-byte word or record size header
        DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) I4TMP(I)
        END DO
        IBLKSZ = I4W
        IOFF=IOFF+(I-1)

C End-of-File (EOF) mark, continue
        IF (IBLKSZ .EQ. 0) GOTO 10

C Next read the block of data
        DO I=1,IBLKSZ
          READ (1, REC=IOFF+I, IOSTAT=IOS) BUFF(I)
          IF (IOS .NE. 0) THEN
            PRINT '("ERROR: BUFF ",I4,X,I4,", IOSTAT: ",I6)', N,I-1,IOS
            IBLKSZ = I-1
            GOTO 50
          END IF
        END DO
        IOFF=IOFF+(I-1)
        N=N+1

C       NOPS Header
        IF (IBLKSZ.EQ.630) THEN
          DO I=1,630
            NOPSHD(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))
          END DO
          PRINT '("NOPS HEADER:",/,A)', NOPSHD(1:630)
          PRINT '("=================")'
          GO TO 50
        END IF

C       Data Buffer Stored as 2-Byte Words
        DO I=1,IBLKSZ/2
          I2TMP = BUFF(2*I-1:2*I)
          I2WORD(I) = I2W
        END DO

        IBLKNO = ISHFT(IAND(I2WORD(1),'0FFF'Z),4) +
     +           ISHFT(I2WORD(1),-12)                          ! Block #
        IBLKID = I2WORD(2)                                     ! Block ID
        IFILNO = I2WORD(3)                                     ! File # on Tape

        IOFFST = 6                                             ! Byte offset
        IF      (IBLKID.EQ.1) THEN                             ! Tape Header
          PRINT '("TAPE HEADER:",3(X,I8))', IBLKNO, IBLKID, IFILNO
          DO I=1,IBLKSZ-IOFFST-2
            STRBUF(I:I) = BUFF(I+IOFFST)
          END DO
          PRINT '(A)', STRBUF(1:80)                            ! String = 80 char
          PRINT '("=================")'
```

```
        ELSE IF (IBLKID.EQ.2) THEN
          PRINT '("FILE HEADER:",3(X,I8))', IBLKNO, IBLKID, IFILNO
          DO WHILE (ICHAR(BUFF(IOFFST+1)).NE.0)
            I2TMP  = BUFF(IOFFST+1:IOFFSET+2)
            IRECSZ = I2W                          ! Length of Record in Bytes
            I2TMP  = BUFF(IOFFST+3:IOFFSET+4)
            IRECNO = I2W                          ! Record Serial Number
            IF (IRECSZ.GT.4) THEN
              DO I=1,IRECSZ-4
                STRBUF(I:I) = BUFF(I+IOFFST+4)
              END DO
              PRINT '(A)', STRBUF(1:IRECSZ-4)
            END IF
            IOFFST = IOFFST+IRECSZ
          END DO
          PRINT '("================")'
        ELSE IF (IBLKID.EQ.3) THEN
          PRINT '("FILE RECORD:",3(X,I8))', IBLKNO, IBLKID, IFILNO
          DO WHILE (I2WORD(IOFFST/2+1).NE.0)
            IRECSZ = I2WORD(IOFFST/2+1)           ! Length of Record in Bytes
            IRECNO = I2WORD(IOFFST/2+2)           ! Record Serial Number
            IRECTP = I2WORD(IOFFST/2+3)           ! Record Type
            IF ((IOFFST+IRECSZ).LE.IBLKSZ) THEN
              IF      (IRECTP.EQ.7200) THEN
                CALL PR7200(I2WORD(IOFFST/2+1:(IOFFST+IRECSZ)/2))
              ELSE IF (IRECTP.EQ.7201) THEN
                CALL PR7201(I2WORD(IOFFST/2+1:(IOFFST+IRECSZ)/2))
              ELSE IF (IRECTP.EQ.7202) THEN
                CALL PR7202(I2WORD(IOFFST/2+1:(IOFFST+IRECSZ)/2))
              ELSE IF (IRECTP.EQ.7203) THEN
                CALL PR7203(I2WORD(IOFFST/2+1:(IOFFST+IRECSZ)/2))
              ELSE
                PRINT '("UNKNOWN TYPE:",3(X,I8))',IRECSZ,IRECNO,IRECTP
                PRINT '("================")'
              END IF
            END IF
            IOFFST = IOFFST + IRECSZ
            IF (IOFFST.GE.IBLKSZ) EXIT
          END DO
        ELSE IF (IBLKID.EQ.132) THEN
          PRINT '("LAST RECORD:",3(X,I8))', IBLKNO, IBLKID, IFILNO
          PRINT '("================")'
        ELSE IF (IBLKID.EQ.196) THEN
          PRINT '("END OF FILE:",3(X,I8))', IBLKNO, IBLKID, IFILNO
          PRINT '("================")'
        ELSE
          PRINT '("UNKNOWN REC:",3(X,I8))', IBLKNO, IBLKID, IFILNO
          PRINT '("================")'
        END IF

C Finally read the last 4-byte word (should match first record size)
   50   DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) I4TMP(I)
        END DO
        IF (IBLKSZ .NE. I4W) THEN
          PRINT '("WARNING: IBLKSZ ",I10," != ",I10)', IBLKSZ, I4W
        ENDIF
```

```
        IOFF=IOFF+(I-1)
      END DO

C Close the input file
   90 CLOSE(1)
      GOTO 100

   99 PRINT '("ERROR: OPEN FILE, IOSTAT: ",I6)', IOS

  100 STOP
      END

C---------------------------------------------------------------------
C     This Subroutine will print the Record Type 7200
C---------------------------------------------------------------------
      SUBROUTINE PR7200(I2WORD)

      INTEGER*2 I2WORD(11)                              ! ARRAY OF 2-BYTE WORDS

      PRINT '("RECLEN =",1(X,I8  ))', I2WORD(1)         ! RECORD LENGTH IN BYTES
      PRINT '("RECNUM =",1(X,I8  ))', I2WORD(2)         ! RECORD NUMBER
      PRINT '("RECID  =",1(X,I8  ))', I2WORD(3)         ! IDENTIFIER (7202)
      PRINT '("-----------------")'
      PRINT '("FILENO =",1(X,I8  ))', I2WORD(4)         ! FILE NUMBER ON TAPE
      PRINT '("YEAR   =",1(X,I8  ))', I2WORD(5)         ! YEAR
      PRINT '("DAY    =",1(X,I8  ))', I2WORD(6)         ! DAY OF YEAR
      PRINT '("TYPE   =",4(X,I8  ))', I2WORD(7:10)      ! RECORD TYPES IN FILE
      PRINT '("-----------------")'
      PRINT '("CHKSUM =",1(X,I8  ))', I2WORD(11)        ! CHECK SUM
      PRINT '("=================")'

      RETURN
      END

C---------------------------------------------------------------------
C     This Subroutine will print the Record Type 7201
C---------------------------------------------------------------------
      SUBROUTINE PR7201(I2WORD)

      INTEGER*2 I2WORD(260)                             ! ARRAY OF 2-BYTE WORDS

      PRINT '("RECLEN =",1(X,I8  ))', I2WORD(1)         ! RECORD LENGTH IN BYTES
      PRINT '("RECNUM =",1(X,I8  ))', I2WORD(2)         ! RECORD NUMBER
      PRINT '("RECID  =",1(X,I8  ))', I2WORD(3)         ! IDENTIFIER (7202)
      PRINT '("-----------------")'
      PRINT '("SYSFLG =",4(X,I8  ))', I2WORD(4:7)       ! SYSTEM FLAGS
      PRINT '("HDRNUM =",1(X,I8  ))', I2WORD(8)         ! HEADER NUMBER
      PRINT '("RESRVD =",5(X,I8  ))', I2WORD(9:13)      ! RESERVED FOR SYSTEM
      PRINT '("ORBRCV =",1(X,I8  ))', I2WORD(14)        ! ORBIT # (AS RECEIVED)
      PRINT '("SEGRCV =",1(X,I8  ))', I2WORD(15)        ! SEGMENT # (AS RECEIVED)
      PRINT '("ORBNUM =",1(X,I8  ))', I2WORD(16)        ! TRUE ORBIT NUMBER
      PRINT '("STDATA =",3(X,I8  ))', I2WORD(17:18),    ! START OF DATA YEAR,DAY,
     +    ISHFT(INT(I2WORD(20)),16) + I2WORD(19)        !              TIME (SEC)
      PRINT '("ENDATA =",3(X,I8  ))', I2WORD(21:22),    ! END OF DATA   YEAR,DAY,
     +    ISHFT(INT(I2WORD(24)),16) + I2WORD(23)        !              TIME (SEC)
      PRINT '("PRDATA =",3(X,I8  ))', I2WORD(25:26),    ! PROD OF DATA  YEAR,DAY,
     +    ISHFT(INT(I2WORD(28)),16) + I2WORD(27)        !              TIME (SEC)
```

```
 PRINT '("ASCEQX =",2(X,I8),X,F8.2)', I2WORD(29), ! ASCENDING EQ XING  DAY,
+    ISHFT(INT(I2WORD(31)),16) + I2WORD(30),      !             TIME (SEC),
+                              I2WORD(32)/100.  !              LON (x100)
 PRINT '("DSCEQX =",2(X,I8),X,F8.2)', I2WORD(33), ! DESCENDING EQ XING DAY,
+    ISHFT(INT(I2WORD(35)),16) + I2WORD(34),      !             TIME (SEC),
+                              I2WORD(36)/100.  !              LON (x100)
 PRINT '("DAYNTX =",2(X,I8),X,F8.2)', I2WORD(37), ! DAY/NIGHT XING     DAY,
+    ISHFT(INT(I2WORD(39)),16) + I2WORD(38),      !             TIME (SEC),
+                              I2WORD(40)/100.  !              LAT (x100)
 PRINT '("NTDAYX =",2(X,I8),X,F8.2)', I2WORD(41), ! NIGHT/DAY XING     DAY,
+    ISHFT(INT(I2WORD(43)),16) + I2WORD(42),      !             TIME (SEC),
+                              I2WORD(44)/100.  !              LAT (x100)
 PRINT '("NMAJFR =",1(X,I8  ))', I2WORD(45)       ! NUMBER OF MAJOR FRAMES
 PRINT '("NCKSXM =",1(X,I8  ))', I2WORD(46)       ! NUM CHKSUM ERRS ON XMIT
 PRINT '("NCKSTP =",1(X,I8  ))', I2WORD(47)       ! NUM CHKSUM ERRS ON TAPE
 PRINT '("NSYNER =",1(X,I8  ))', I2WORD(48)       ! NUM SYNC ERRORS
 PRINT '("NRDOUT =",1(X,I8  ))', I2WORD(49)       ! NUM VERIFICATION RDOUTS
 PRINT '("NLNCDA =",1(X,I8  ))', I2WORD(50)       ! NUM LENGTH OF CDA FILE
 PRINT '("STMODE =",2(X,I8  ))', I2WORD(51:52)    ! MODE AT START OF DAY
 PRINT '("ENMODE =",2(X,I8  ))', I2WORD(53:54)    ! MODE AT END OF DAY
 PRINT '("ERRFLG =",1(X,I8  ))', I2WORD(55)       ! LAT/LON ERROR FLAG
 PRINT '("NCOEFS =",1(X,I8  ))', I2WORD(56)       ! NUM EIGEN COEF USED
 PRINT '("NTLVLS =",1(X,I8  ))', I2WORD(57)       ! NUM TEMP SUBLEVELS USED
 PRINT '("YRBLK  =",1(X,I8  ))', I2WORD(58)       ! YEAR OF SUB BLOCK USED
 PRINT '("DAYBLK =",1(X,I8  ))', I2WORD(59)       ! DAY OF SUB BLOCK USED
 PRINT '("TPMC   =",7(X,F8.2))', I2WORD(60:66)/100.!PMC MEAN TEMP (DEGx100)
 PRINT '("PRPMC  =",7(X,F8.2))', I2WORD(67:73)/100.!PMC MEAN PRES (MB x100)
 PRINT '("PPMC   =",7(X,I8  ))', I2WORD(74:80)    ! PMC MEAN PERIOD(COUNTS)
 PRINT '("FLGLIM =",1(X,I8  ))', I2WORD(81)       ! FLAG DATA LIMITED PROC
 PRINT '("SPARE1 =",8(X,I8  ),16(/,8X,8(X,I8)))',
+                              I2WORD(82:212)   ! SPARES
 PRINT '("VERPRG =",1(X,F8.1))', I2WORD(213)/10.  ! VERSION NUM PROG (x10)
 PRINT '("VERFMT =",1(X,I8  ))', I2WORD(214)      ! VERSION NUM FORMAT
 PRINT '("SPARE2 =",8(X,I8  ),5(/,8X,8(X,I8)))',
+                              I2WORD(215:259)  ! SPARES
 PRINT '("-----------------")'
 PRINT '("CHKSUM =",1(X,I8  ))', I2WORD(260)      ! CHECK SUM
 PRINT '("=================")'

 RETURN
 END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the Record Type 7202
C-----------------------------------------------------------------------
      SUBROUTINE PR7202(I2WORD)

      INTEGER*2     I2WORD(388)                         ! ARRAY OF 2-BYTE WORDS
      CHARACTER     FLAGS(2,3)
      CHARACTER     DIGB(2,6)
      CHARACTER     CHANID(4,9)
      INTEGER*2     I2TMP                               ! 15 word data record
      INTEGER*4     I4TMP                               ! 15 word data record
      REAL*4        CALRAD(8,12)                        ! Calib 8 words x 12 slots
      CHARACTER     TMP2(2)                             ! Temp buffer for data
      CHARACTER     TMP4(4)                             ! Temp buffer for data
      EQUIVALENCE   (TMP2,I2TMP)
      EQUIVALENCE   (TMP4,I4TMP)

      PRINT '("RECLEN =",1(X,I8  ))', I2WORD(1)        ! RECORD LENGTH IN BYTES
      PRINT '("RECNUM =",1(X,I8  ))', I2WORD(2)        ! RECORD NUMBER
      PRINT '("RECID  =",1(X,I8  ))', I2WORD(3)        ! IDENTIFIER (7202)
      PRINT '("-----------------")'
      PRINT '("FMTMRK =",2(X,I8  ))',                  ! FORMAT/MARK
     +  IAND(I2WORD(4),'FF'Z), ISHFT(I2WORD(4),-8)
      PRINT '("ERRFLG =",2(X,B8.8))',                  ! ERROR FLAGS (SEE DOC)
     +  ISHFT(I2WORD(5),-8), IAND(I2WORD(5),'FF'Z)
      PRINT '("YEAR   =",1(X,I8  ))', I2WORD(6)        ! YEAR
      PRINT '("DAY    =",1(X,I8  ))', I2WORD(7)        ! DAY OF YEAR
      PRINT '("TIME   =",1(X,I8  ))',                  ! TIME (SECONDS)
     +  IOR(ISHFT(INT(I2WORD(9)),16),IAND(INT(I2WORD(8)),'FFFF'Z))
      PRINT '("LATSAT =",1(X,F8.2))', I2WORD(10)/100.  ! SAT LATITUDE (x100 DEG)
      PRINT '("LONSAT =",1(X,F8.2))', I2WORD(11)/100.  ! SAT LONGITUDE(x100 DEG)
      PRINT '("ALTSAT =",1(X,I8  ))', I2WORD(12)       ! SAT ALTITUDE (KM)
      PRINT '("LATTP  =",1(X,F8.2))', I2WORD(13)/100.  ! LAT OF TAN PT(x100 DEG)
      PRINT '("LONTP  =",1(X,F8.2))', I2WORD(14)/100.  ! LON OF TAN PT(x100 DEG)
      DO I = 1,3                                       ! FLAGS
        I2TMP = I2WORD(I-1+15)                         ! (SEE DOC FOR VALUES)
        FLAGS(1:2,I) = TMP2
      END DO
      PRINT '("FLAGS  =",2(X,B8.8),2(/,8X,2(X,B8.8)))',! REVERSED FOR DISPLAY
     +  ((FLAGS(J,I), J=2,1,-1), I=1,3)
      PRINT '("TBBODY =",1(X,F8.2))', I2WORD(18)/100.  ! BLACK BODY TEMP(C x100)
      PRINT '("TCHOP  =",1(X,F8.2))', I2WORD(19)/100.  ! CHOPPER TEMP(DEG Cx100)
      DO I = 1,6                                       ! DIGITAL B
        I2TMP = I2WORD(I-1+20)                         ! (SEE DOC FOR VALUES)
        DIGB(1:2,I) = TMP2
      END DO
      PRINT '("DIGB   =",2(X,B8.8),5(/,8X,2(X,B8.8)))',! REVERSED FOR DISPLAY
     +  ((DIGB(J,I), J=2,1,-1), I=1,6)
      PRINT '("SCSTAT =",5(X,I8  ))', I2WORD(26:30)    ! SPACECRAFT STATUS
      DO I = 1,18,2                                    ! CHANNEL ID
        I4TMP = IOR(ISHFT(INT(I2WORD(I-1+32)),16),     ! (SEE DOC FOR VALUES)
     +          IAND(INT(I2WORD(I-1+31)),'FFFF'Z))
        CHANID(1:4,(I+1)/2) = TMP4
      END DO
      PRINT '("CHANID =",4(X,B8.8),8(/,8X,4(X,B8.8)))',! REVERSED FOR DISPLAY
+  ((CHANID(J,I), J=4,1,-1), I=1,9)
```

```
 PRINT '("CALRAD =",8(X,I8  ),11(/,8X,8(X,I8)))', ! RADIANCES(8 WORDS x 12)
+                                  I2WORD(49:144)   ! (SEE DOC FOR SCALING)
 PRINT '("RADERR =",8(X,I8  ),/,8X,4(X,I8))',
+                                  I2WORD(145:156)  ! RAD EST.ERRORS(12 WORD)
 PRINT '("PRPMC  =",7(X,I8  ))', I2WORD(157:163)  ! PMC PRESSURES
 PRINT '("TPMC   =",7(X,F8.2))', I2WORD(164:170)/100. ! PMC TEMP (C x100)
 PRINT '("AZIM   =",8(X,F8.3))', I2WORD(171:178)/1000.! AZIM POS(DEG x1000)
 PRINT '("LVDT   =",8(X,F8.3))', I2WORD(179:186)/1000.! LVDT (DEG x1000)
 PRINT '("ROLLD  =",8(X,F8.3))', I2WORD(187:194)/1000.! ROLL CH.D(DEGx1000)
 PRINT '("TANHTD =",8(X,F8.2))', I2WORD(195:202)/100. ! TAN HT CH.D(KMx100)
 PRINT '("THSCLA =",8(X,F8.3))', I2WORD(203:210)/1000.! TH SCL CH.A (x1000)
 PRINT '("THSCLB =",8(X,F8.3))', I2WORD(211:218)/1000.! TH SCL CH.B (x1000)
 PRINT '("THSCLC =",8(X,F8.3))', I2WORD(219:226)/1000.! TH SCL CH.C (x1000)
 PRINT '("ZTERR  =",8(X,F8.3))', I2WORD(227:234)/1000.! EST. ERR ZT (x1000)
 PRINT '("ROLLDR =",8(X,F8.4))', I2WORD(235:242)/10000.!ROLL DERIV (x10000)
 PRINT '("RERRDR =",8(X,F8.4))', I2WORD(243:250)/10000.!ERROR IN ABOVE
 PRINT '("RACSER =",8(X,I8  ))', I2WORD(251:258)  ! ACS ROLL FINE ERROR
 PRINT '("TERMHT =",1(X,F8.2))', I2WORD(259)/100. ! TERM HT TAN PT(KM x100)
 PRINT '("YAW    =",8(X,I8  ))', I2WORD(260:267)  ! YAW
 PRINT '("PITCH  =",8(X,I8  ))', I2WORD(268:275)  ! PITCH
 PRINT '("URADA  =",8(X,I8  ),3(/,8X,8(X,I8)))',  ! 4x8
+                                  I2WORD(276:307)  ! UNCAL RAD(A1P/W,A234P/W)
 PRINT '("URADBC =",8(X,I8  ),5(/,8X,8(X,I8)))',  ! 6x8
+                                  I2WORD(308:355)  ! UNCAL(B2P/W,C1P/W,C23P/W
 PRINT '("PPMC   =",7(X,I8  ))', I2WORD(356:362)  ! PMC PERIODS
 PRINT '("APMC   =",7(X,I8  ))', I2WORD(363:369)  ! PMC AMPLITUDES
 PRINT '("ACHOP  =",1(X,I8  ))', I2WORD(370)      ! CHOPPER AMPLITUDE
 PRINT '("TSIEVE =",7(X,I8  ))', I2WORD(371:377)  ! SIEVE TEMPERATURES
 PRINT '("TCHOPX =",1(X,I8  ))', I2WORD(378)      ! CHOPPER TEMP EXPANDED
 PRINT '("TDETEC =",4(X,I8  ))', I2WORD(379:382)  ! A234,B1,A1,C1 DETEC T
 PRINT '("DCLVL  =",2(X,I8  ))', I2WORD(383:384)  ! B1,A234 DETEC DC LEVEL
 PRINT '("TMIRR  =",2(X,I8  ))', I2WORD(385:386)  ! M1,M2 MIRROR TEMP
 PRINT '("TSEU   =",1(X,I8  ))', I2WORD(387)      ! SEU TEMPERATURE 'A'
 PRINT '("-----------------")'
 PRINT '("CHKSUM =",1(X,I8  ))', I2WORD(388)      ! CHECK SUM
 PRINT '("=================")'

 RETURN
 END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the Record Type 7203
C-----------------------------------------------------------------------
      SUBROUTINE PR7203(I2WORD)

      INTEGER*2 I2WORD(388)                              ! ARRAY OF 2-BYTE WORDS

      NOE=8
      NR=10
      PRINT '("RECLEN =",1(X,I8  ))', I2WORD(1)        ! RECORD LENGTH IN BYTES
      PRINT '("RECNUM =",1(X,I8  ))', I2WORD(2)        ! RECORD NUMBER
                                                       ! ----------------------
      PRINT '("RECID  =",1(X,I8  ))', I2WORD(3)        ! IDENTIFIER (7203)
      N=0
      DO I=1,3
        PRINT '("-----------------")'
        PRINT '("LATREF =",1(X,I8  ))', I2WORD(N+4)    ! LAT REF(1=50S,48=67.5N)
        PRINT '("TIME   =",1(X,I8  ))',
     +  ISHFT(INT(I2WORD(6)),16) + I2WORD(5)           ! TIME (SECONDS)
        PRINT '("DAY    =",1(X,I8  ))', I2WORD(N+7)    ! DAY OF YEAR
        PRINT '("YEAR   =",1(X,I8  ))', I2WORD(N+8)    ! YEAR
        PRINT '("LATSAT =",1(X,F8.2))', I2WORD(N+9)/100. ! SAT LAT (x100 DEG)
        PRINT '("LONSAT =",1(X,F8.2))', I2WORD(N+10)/100.! SAT LON (x100 DEG)
        PRINT '("LATTP  =",1(X,F8.2))', I2WORD(N+11)/100.! TAN PT LAT (x100 DEG)
        PRINT '("LONTP  =",1(X,F8.2))', I2WORD(N+12)/100.! TAN PT LON (x100 DEG)
        PRINT '("CALMF  =",1(X,I8  ))', I2WORD(N+13)   ! MAJOR FRAME IN CAL FILE
        PRINT '("COEF   =",8(X,F8.4))',                ! EIGEN FUNC COEFS x10000
     +  I2WORD(N+14+0*NOE:N+14+1*NOE-1)/10000.
        PRINT '("COEFSD =",8(X,F8.4))',                ! STD DEV OF ABOVE x10000
     +  I2WORD(N+14+1*NOE:N+14+2*NOE-1)/10000.
        PRINT '("TEMP   =",8(X,F8.1),1(/,8X,2(X,F8.1)))',
     +  I2WORD(N+14+2*NOE+0*NR:N+14+2*NOE+1*NR-1)/10.  ! TEMP(K)AT EACH LEVELx10
        PRINT '("TEMPRE =",8(X,F8.4),1(/,8X,2(X,F8.4)))',
     +  I2WORD(N+14+2*NOE+1*NR:N+14+2*NOE+2*NR-1)/10000.!REL ERROR ABOVE x10000
        PRINT '("TEMPSD =",8(X,F8.1),1(/,8X,2(X,F8.1)))',
     +  I2WORD(N+14+2*NOE+2*NR:N+14+2*NOE+3*NR-1)/10.  ! STD DEV OF TEMP x10
        PRINT '("TEMPAB =",8(X,F8.1),8(/,8X,8(X,F8.1)))',
     +  I2WORD(N+14+2*NOE+3*NR:N+14+2*NOE+3*NR+72-1)/10.!ABS(T)x10 0.2INC LOG(P)
        N=N+128
      END DO
      PRINT '("-----------------")'
      PRINT '("CHKSUM =",10(X,I8  ))', I2WORD(388)     ! CHECKSUM
      PRINT '("=================")'

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will convert 24-bit word to 32-bits
C-----------------------------------------------------------------------
      SUBROUTINE I24I32(IWORD)

      INTEGER*4     I4TEMP                      ! 15 word data record
      CHARACTER     TEMP(4)                     ! Temp buffer for data record
      EQUIVALENCE   (TEMP,I4TEMP)

      I4TEMP = IWORD
      IW = 0
      DO K = 1,4
        IW = ISHFT(IW, 6)                       ! Shift left by 6 bits
        IW = IOR(IW,IAND(ICHAR(TEMP(K)),Z'3F')) ! Remove 2 most signif bits
      END DO
      IF (ISHFT(IW,-23).EQ.1) THEN
        IW = IW - 2**24                         ! Negative value
      END IF

      IWORD = IW

      RETURN
      END

C-----------------------------------------------------------------------
C     This Function returns EBCDIC to ASCII character index (255=nothing)
C-----------------------------------------------------------------------
      FUNCTION IEBC(I)

      INTEGER   EBCTBL(256)

      DATA EBCTBL /
     + 000,001,002,003,255,009,255,127,255,255,255,011,012,013,014,015, ! 0_
     + 016,017,018,019,255,133,008,255,024,025,255,255,028,029,030,031, ! 1_
     + 255,255,255,255,255,010,023,027,255,255,255,255,255,005,006,007, ! 2_
     + 255,255,022,255,255,255,255,004,255,255,255,255,020,021,255,026, ! 3_
     + 032,255,255,255,255,255,255,255,255,255,162,046,060,040,043,124, ! 4_
     + 038,255,255,255,255,255,255,255,255,255,033,036,042,041,059,172, ! 5_
     + 045,047,255,255,255,255,255,255,255,255,166,044,037,095,062,063, ! 6_
     + 255,255,255,255,255,255,255,255,255,096,058,035,064,039,061,034, ! 7_
     + 255,097,098,099,100,101,102,103,104,105,255,255,255,255,255,177, ! 8_
     + 255,106,107,108,109,110,111,112,113,114,255,255,255,255,255,255, ! 9_
     + 255,126,115,116,117,118,119,120,121,122,255,255,255,255,255,255, ! a_
     + 094,255,255,255,255,255,255,255,255,255,091,093,255,255,255,255, ! b_
     + 123,065,066,067,068,069,070,071,072,073,255,255,255,255,255,255, ! c_
     + 125,074,075,076,077,078,079,080,081,082,255,255,255,255,255,255, ! d_
     + 092,255,083,084,085,086,087,088,089,090,255,255,255,255,255,255, ! e_
     + 048,049,050,051,052,053,054,055,056,057,255,255,255,255,255,159/ ! f_
C       _0  _1  _2  _3  _4  _5  _6  _7  _8  _9  _A  _B  _C  _D  _E  _F

      IEBC = EBCTBL(I+1)

      RETURN
      END
```