A COMPUTER ORIENTED SYSTEM FOR LOGGING, ANNOTATING, TABULATING AND PLOTTING MARINE GEOPHYSICAL AND GEOLOGICAL DATA COLLECTED BY THE PACIFIC-ARCTIC BRANCH OF THE OFFICE OF MARINE GEOLOGY, U.S. GEOLOGICAL SURVEY

BY

BRADLEY R. LARSEN, ALAN K. COOPER, GRAIG A. McHENDRIE, AND JAMES V. GARDNER

U.S. GEOLOGICAL SURVEY

OPEN FILE REPORT

NO. 78-288

This report is preliminary and has not been edited or reviewed for conformity with Geological Survey standards.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

INTRODUCTION

In late 1974, the Pacific-Arctic branch of the Office of Marine Geology, U.S. Geological Survey started developing a computer-oriented system for logging, annotating, and tabulating the large quantity of oceanographic data routinely collected in the 8 project areas off the west coast of North America (see fig. 1). The system, which is now used in both shipboard and land based data management operations, is fully described in this report. Organizationally, this system is part of a larger data management project (Marine Data Information Group), under the direction of Tom Chase, that also provides for the preservation, archiving and distribution of all marine data. The shipboard procedures, logging forms, computer programs and indexing schemes described herein effectively outline the mechanics of how the Office of Marine Geology can accurately index, plot, locate, and recover data collected on its cruises.

The data organization system is based on a concept that is currently used at Scripps Institute of Oceanography, namely that a hierarchal canopy of code letters ("ladder codes") can be used to uniquely identify any oceanographic data. Once the appropriate "ladder code" (or related mnemonic code) identifying the data, and the time at which the data were collected are known, the computer can produce a multitude of organized data listings and data plots, as required.

The Media and Data Codes (mnemonic versions of the 2nd and 4th levels of the "ladder codes") as well as the data collection times are entered into the shipboard logs during the cruise. After the cruise, a master index of coded information, the Cruise Data Index (CDI), is generated from the shipboard logs by the computer. Three items are usually produced from the CDI: 1. A Cruise Report that briefly describes how much data was collected; 2. A Cruise Data listing that categorically lists in detail all data collected on the cruise; 3. An optional set of data plots such as sample location maps, geophysical data trackline maps, etc.

The first section of the manual is addressed to the person or persons (Cruise Curator) responsible for setting up this system on a cruise. It outlines his or her responsibilities and the necessary steps which must be followed to ensure that the system works on-board ship.

Section two outlines the actual data logging and annotating requirements and procedures of the system and is prefaced by a section orientating watchstanders to their duties and defining special terms and policy used by the system. It also contains examples of how to use data labels, how to annotate records, and how to fill out each type of logsheet.

The third and fourth sections provide the details on using the special logging forms related to navigation, geologic sample description, sonobuoys, and gravity land ties. Examples are provided.

The fifth section contains a listing of the mnemonic Media and Data Codes which are used on the shipboard logforms to identify specific data types. The section also contains diagrams and cross-referenced listings of the "ladder codes" which are assigned internally in the computer to provide the most complete identification of the data in a Cruise Data Index.

The final section of the manual contains the actual computer software that has been created to manipulate Cruise Data Index files. Documentation and examples have been provided to explain how to run the programs and how the programs work. These programs are EDITCDI, STATCDI, and PLOTCDI.

ACKNOWLEDGEMENTS

This report is the result of much trial and effort over a 3 year period to iron out a working data inventorying/plotting system for this Branch. We are grateful to the many people who have helped us clarify our original concepts of how this should be done and who have helped us to keep focused on the essentials of the task. Special acknowledgements and thanks are due to Tom Chase for his backing and encouragement as well as to Paula Quinterno who gave much of her time and effort in reviewing and critiquing this report.

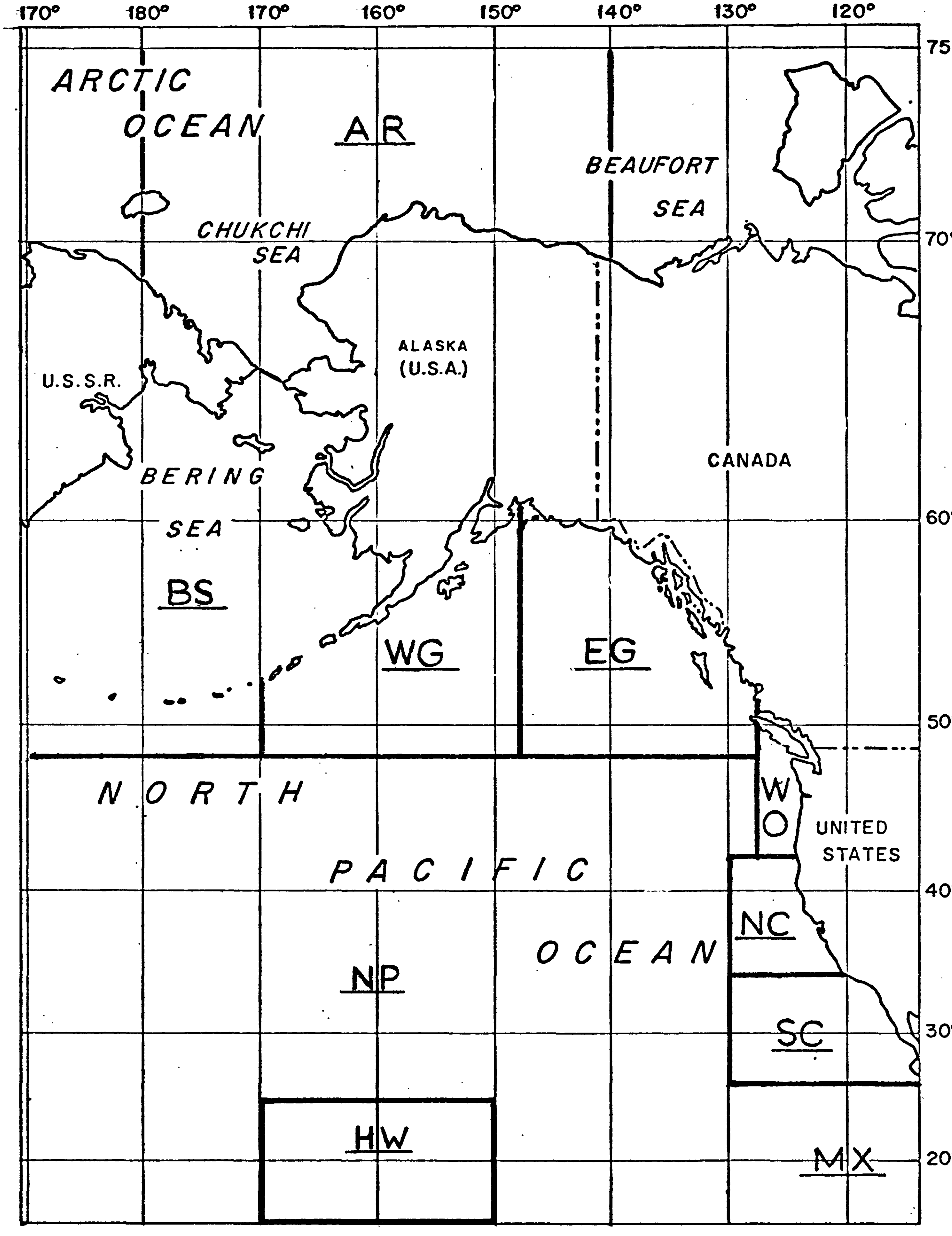


FIGURE 1 AREA CODES USED TO IDENTIFY PROJECT AREAS

SECTION ONE

The Cruise Curator is a person designated by the Chief Scientist of a cruise or by the Marine Data Information Group to oversee the operation of the data logging, annotating, and monitoring system used on-board cruises of the Office of Marine Geology. The Cruise Curator's general responsibilities are to ensure that:

a) Sufficient supplies for logging and annotating data are on-board ship.

b) Proper procedures for logging, annotating, indexing and labeling data are followed by scientific personnel.

c) All samples and records are brought safely back to the Branch.

This section details the responsibilities of a Cruise Curator and gives general guidelines for carrying out his or her duties in an effective manner.

RESPONSIBILITIES OF THE CRUISE CURATOR

-Outline-

I. PRE-CRUISE

A. Become familiar with MDIG procedures

B. Ensure that sufficient logging materials are aboard ship

II. DURING CRUISE

A. Overall responsibilities

B. Immediate duties - first day

C. Daily duties

D. Cruise Data Index - special entries

III. POST-CRUISE

A. Immediate duties - first day

B. Duties at home - USGS

IV. CRUISE CURATOR'S CHECKLIST

RESPONSIBILITIES OF THE CRUISE CURATOR

I. PRE-CRUISE

A. Become familiar with the established procedures for data handling during all phases of the cruise

1. Read all documentation but concentrate on the Cruise Curator's Checklist on page 10 and the Minimum Marine Data/Logging Requirements for Geophysical Systems on pages 13 and 14.

2. Discuss procedures with MDIG Curator and Chief Scientists

3. Help instruct cruise members in logging and annotating procedures

4. Obtain "Cruise Locator" from MDIG Coordinator

B. Gather Watchstanders' supplies together or ensure that there will be sufficient supplies aboard ship

1. Watchstanders' supplies include:

a. Rubber stamps and pads

b. Logbook binders and blank forms

c. Labels

2. For S.P. Lee and Sea Sounder cruises, supplement existing supplies aboard ship, if necessary

II. DURING CRUISE

A. Overall Responsibility for:

1. Accuracy, clarity and completeness of all:

a. Logbook entries

b. Analog record annotations (analog record is a paper roll containing continuous-trace data)

c. Physical data labels

2. Proper storage aboard ship for:

a. Completed logbook forms

b. Analog and digital records

c. Physical samples (cores, dredges, water samples, etc.)

d. Miscellaneous physical cruise data (maps, plots, etc.)

3. Proper transferral of cruise data from ship to USGS

B. Immediate duties - first day

1. Organize, supply, or set up

a. Logbooks for all watch and instrument locations

1) In general, set up one logbook for each instrument. If necessary, a logbook can contain the logsheets for more than one instrument type; in this case, partition the logbook into more than one section

2) The Navigation Logbook (non-contract navigation cruises) should also contain a section for Cruise Data Index logsheets, so navigators can record start/ end of tracklines

b. Specific "How to Log/Annotate" diagrams with entries for each underway data system. See pages 13, 14, and 36.

2. Assist and instruct underway watchstanders in procedures for:

a. Logbook entries

b. Record annotation

c. Physical data labels

The chief scientist should appoint one person from each watch to help instruct watchstanders in these procedures and to help the Cruise Curator check that they are properly followed.

C. Daily duties

1. Collect all completed logbook sheets except for the last one or two and:

a. Check them for accuracy, clarity, and completeness

b. Check all entries that are to be keypunched from the collected logsheets (CDI, Station Operations, Acoustic and Geopotential Equipment logs) to ensure their completeness and that:

1) All the "required" entries have been entered (see GENERAL DEFINITIONS AND POLICY FOR DATA RECORDING, part II. F., page 21 for a list of the "required" entries to be keypunched).

2) Every "start" or "on" entry has a corresponding "end" or "off" entry

3) Correct Media Codes and Archive Numbers have been assigned

c. Place the logbook sheets into a single binder for storage

2. Collect all completed analog rolls and magnetic tape and:
    a. Check for accuracy, clarity and completeness of:
        1) Labels on rolls or reels and their storage boxes
        2) Rubber stamp or label entries at: (see page 36)
            a. Start/end of roll
            b. Start/end of line (spot check for this)
            c. Equipment failures (spot check for this)
    b. Place in proper storage location
3. Collect all sample and other physical cruise data and:
    a. Check for accuracy, clarity and completeness of labels
    b. Place in proper storage location

D. Cruise Data Index (CDI)

1. Ensure that "required" entries (not done by the watchstander) are made for:
    a. Inventory of major scientific gear (Sat-Nav., 160 KJ Seismic, piston core, etc.) to be used on the cruise
    b. Roster of:
        1) Scientific personnel (include when they came on board ship for duty and when they left - use the on/off columns
        2) Ships key personnel (Capt., Chief Engineer, Chief Mate)
    c. List of time and places for:
        1) Port stops
        2) Start/end of cruise
    d. Miscellaneous physical cruise data
        1) All maps and plots
        2) Other miscellaneous data

III. POST-CRUISE

A. Immediate duties - first day

1. Collect, sort, inventory, and package for shipment or where appropriate stow:
    a. Re-usable logging supples: (listed on MDIG shipboard Inventory Sheet, example on page 15.
        1) Discard all unused supplies which have been imprinted with specific cruise codes (i.e., log sheets, labels, etc.)

2) Discard all wornout materials (i.e., logbook binders, stamp pads, etc.)

3) Inform the Chief Scientist for the next Cruise or Tom Chase (by radiotelephone, etc.) about logging supplies which might be running low.

4) Except for S.P. LEE and SEA SOUNDER cruises, return all usable supplies from your cruise to USGS

5) For S.P. LEE and SEA SOUNDER cruises, inventory and leave all usable logging supplies in appropriate locations aboard ship

b. Recorded data

1) Logbook forms

2) Encoding forms (used for digitized information)

3) Analog rolls and printer lists

4) Magnetic and punch tapes

5) Maps and plots

6) Miscellaneous physical data

c. Physical station data (rock, sediment, water samples, etc.)

2. Make arrangements and supervise the shipping of all cruise data from ship to USGS in Menlo Park, CA and keep a written record of how this is done using SHIP DATA INVENTORY/TRANSFER LOG (see page 15). It is your responsibility to know how every piece of data is to arrive safely in Menlo Park.

B. Duties at home - USGS

1. Assist permanent Marine Data Information staff with:

a. Assembling and making copies of scientific equipment logs

b. Editing all log entries which are to be keypunched

c. Microfilming and reproduction of analog rolls

d. Inventorying and locating all physical data that has been collected on the cruise

IV. CRUISE CURATOR'S CHECKLIST

A. General concerns

1. Logging Procedures - Know minimum requirements for each instrument system. There are two categories of 'necessary' log entries:
   - a. Routine entries at fixed time intervals. Ask: What is logged at what time interval?
   - b. Entries that note change in data being recorded, or in system that records the data: the start/end/on/off of data recording intervals; changes in recorder or system such as scale, sweep, power, etc.
2. Annotating procedures - Know minimum requirements for each instrument system. The categories of 'necessary' annotations are the same as those for 'necessary' log entries. However, an additional requirement is to place time marks on the records.
3. Know logging requirements for the portions of logs which are keypunched to make up a Cruise Data Index. The basic idea is that we have to know both where samples are collected and over what distance interval data is recorded. This entails making sure that the times that samples are actually collected and the time intervals over which data is recorded are flagged for keypunching. Remember, navigation is keyed to time.
4. Monitor the above by periodic checking. Have the Chief Scientist assign 'deputies' (one of the watchmembers) to each watch to help educate the members of the watch in these procedures.
5. Keep track of the data.

B. Practical concerns

1. Know which data-recording systems or sampling devices will be used on your particular cruise.
   - a. Learn where they are aboard ship and what they look like.
   - b. Make a list of their Data Codes, and for data-recording systems, know the different media they will be recording on and the corresponding Media Codes to be used with them.
   - c. Learn which of the systems are recording their data alongside data from other systems on the same magnetic tape or paper roll, etc. The electronic technician on board can help

with this question. Give special attention to learning how these 'multi-instrument' systems are logged.

2. Set up logbooks near recorders with the appropriate logsheets inserted and make sure there are ample marking pencils and pens nearby. If convenient, it is permissible to keep logsheets for separate systems in one binder. Encourage the use of pencils for logging.
3. Post lists of Data/Media Codes to be used near recorders and where sample-logging is done. Also, post a list outlining logging/annotating rules for each instrument type.
4. Check logs once a-day for completeness and correctness, i.e., make sure it is being done right and that sampling times and data recording intervals for each instrument are flagged for keypunching; nip snafus in the bud. Spot check analog records (paper rolls recording continuous trace data); at least make sure roll/tape number sequences for each system are right and occasionally look through individual rolls for completeness.
5. The following are reminders to take care of tasks that have been neglected several times on past cruises:
   a. Remember to log the recording intervals of the multichannel data which is being recorded on magnetic tape. This means when the system goes down in the middle of a tape as well as the start/end of the tape. You can do this by periodically checking the Shot Point Log kept in the multichannel area. It will indicate tape numbers and when data is or is not recording on the tape.
   b. Remember to list all navigational charts and maps generated on your cruise in the Cruise Data Index log; and make sure start/end/on/off times are logged for SAT/NAV printer listings, miniranger tapes, digital magnetic navigation tapes, etc.
   c. Be sure you have also properly logged lists of personnel with on/off entries as well as lists of equipment, port stops and the start and end of the cruise in the CDI log.

d. If gravity data is being collected on your cruise, be sure gravity land ties are made at the beginning and end of the cruise as well as each time the ship comes in to port during the cruise. Read instructions on pages 56 to 59.

e. Finally, be sure you do not neglect the logging and inventorying of <u>any</u> data, whether geologic, geophysical, hydrographic, or navigational, being collected on your cruise.

6. Store the data so you know where it is and can conveniently inventory it. You must make sure you know how much data was collected on the cruise and how it is to be transported back to the Geological Survey in Menlo Park.

MINIMUM MARINE DATA LOGGING/ANNOTATING REQUIREMENTS FOR GEOPHYSICAL SYSTEMS

- LOG ENTRIES -

	Acoustic Equipment Logs	Geopotential Equipment Logs
System / Entry time	All Systems	All Systems
Time of Occurence	At Start/End of data recording on an analog roll, magnetic tape, etc., and also whenever data stops, then starts recording again <u>within</u> an analog roll, magnetic tape, etc.	
Time of Occurence	Changes in recorder scale, sweep rate, shoot rate; changes in equipment filter settings, power available	Changes in number of sensors operating or in sensor towing distance.
Every 4 Hours	Routine entry of all logsheet column headers at start of every 4 hour watch or every 4 hours - to include name of the new watchstander.	

- ANNOTATIONS -

	Acoustic Analog Records	Geopotential Analog Records
System / When to annotate	All Systems	All Systems
Time of Occurence	Start/End of roll labels must be placed at each end of analog record on same side as data.	
Time of Occurence	Start/End of line label must be placed just before/after line changes.	
Time of Occurence	Changes in recording system that affect appearance of data; i.e., changes in scale, filter settings, fire/sweep rate, equipment failure. Time and change must be noted on the record.	
Every Hour	Routine annotation of day, time, recorder scale on every analog record.	

FIGURE 2 MINIMUM LOGGING/ANNOTATING REQUIREMENTS FOR GEOPHYSICAL SYSTEMS

This is an example of how a Cruise Curator might fill out the sheet on page 13 for a particular cruise. The entries are purely arbitrary and should not influence how the sheet is filled out in a real situation. What is logged or annotated and how frequently it is done should be decided by the Chief Scientist. However, the minimum standards outlined on page 13 must be followed.

MINIMUM MARINE DATA LOGGING/ANNOTATING REQUIREMENTS FOR GEOPHYSICAL SYSTEMS

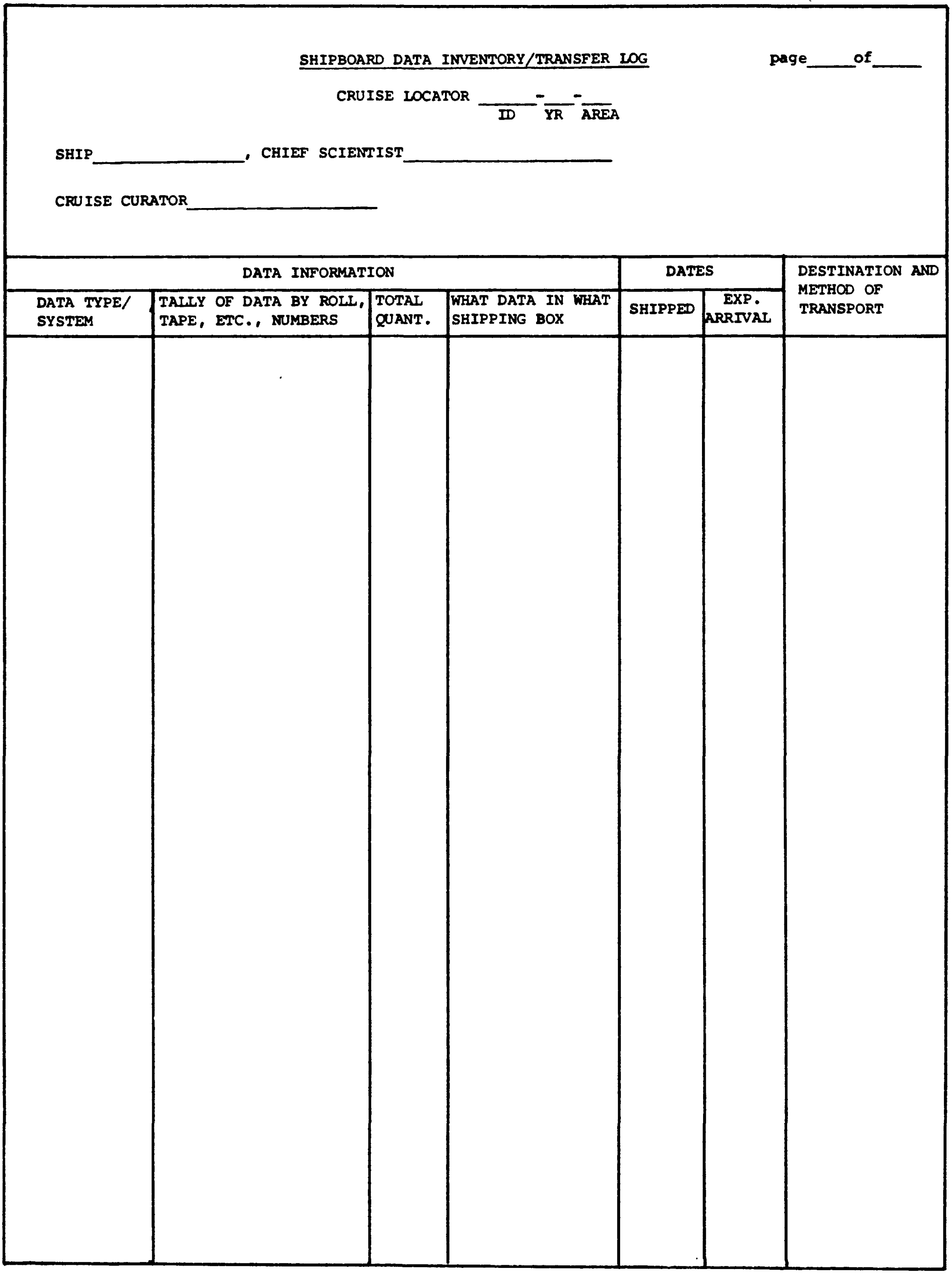
- LOG ENTRIES -

	Acoustic Equipment Logs				Geopotential Equipment Logs	
Entry time / System	160 KJ SPARKER	UNIBOOM	BATHY	SIDE SCAN SONAR	MAGNETOMETER	GRAVITY METER
Time of Occurence	At Start/End of data recording on an analog roll, magnetic tape, etc., and also whenever data stops, then starts recording again within an analog roll, magnetic tape, etc.					
Time of Occurence	Changes in recorder scale, sweep rate, shoot rate; changes in equipment filter settings, power available				Changes in number of sensors operating or in sensor towing distance.	
EVERY HOUR	DAY/TIME, SCALE, DEPTH, FILTERS, COURSE	DAY/TIME, SCALE, SWEEP DEPTH	DAY/TIME, SCALE, SWEEP, COURSE, SPEED	DAY/TIME, SCALE	DAY/TIME, SCALE, SAMPLE RATE, MASTER/SLAVE READINGS	DAY/TIME, SCALE, SAMPLE RATE, DIGITAL & ANALOG READINGS
EVERY ½ HOUR	DAY/TIME DEPTH, COURSE SPEED	DAY/TIME DEPTH	DAY/TIME COURSE, SPEED	DAY/TIME LINE NUMBER	DAY/TIME, SCALE	DAY/TIME
EVERY 10 MINUTES	—	DAY/TIME DEPTH	—	—	—	—

- ANNOTATIONS -

	Acoustic Analog Records				Geopotential Analog Records	
When to Annotate / System	160 KJ SPARKER	UNIBOOM	BATHY	SIDE SCAN SONAR	MAGNETOMETER	GRAVITY METER
Time of Occurence	Start/End of roll labels must be placed at each end of analog record on same side as data.					
Time of Occurence	Start/End of line label must be placed just before/after line changes.					
Time of Occurence	Changes in recording system that affect appearance of data; i.e., changes in scale, filter settings, fire/sweep rate, equipment failure. Time and change must be noted on the record.					
EVERY HOUR	DAY/TIME, SCALE, POWER FILTER SETTINGS	DAY/TIME, SCALE, SWEEP, COURSE, SPEED	DAY/TIME, SCALE, COURSE SPEED	DAY/TIME, SCALE, LINE NUMBER	DAY/TIME, SCALE	DAY/TIME, SCALE
EVERY ½ HOUR	DAY/TIME, COURSE, SPEED, LINE NUMBER	DAY/TIME, COURSE, SPEED, DEPTH	DAY/TIME, COURSE, SPEED, DEPTH	DAY/TIME, LINE NUMBER	—	—
EVERY 15 MINUTES	DAY/TIME	DAY/TIME, DEPTH	—	—	—	—
EVERY 5 MINUTES	—	DAY/TIME	—	—	—	—

FIGURE 3 EXAMPLE OF HOW TO FILL OUT MINIMUM LOGGING/ANNOTATING REQUIREMENTS SHEET ON PAGE 13



SHIPBOARD DATA INVENTORY/TRANSFER LOG

page_____of_____

CRUISE LOCATOR ______-___-___
ID YR AREA

SHIP________________, CHIEF SCIENTIST______________________

CRUISE CURATOR____________________

DATA INFORMATION				DATES		DESTINATION AND METHOD OF TRANSPORT
DATA TYPE/ SYSTEM	TALLY OF DATA BY ROLL, TAPE, ETC., NUMBERS	TOTAL QUANT.	WHAT DATA IN WHAT SHIPPING BOX	SHIPPED	EXP. ARRIVAL	

FIGURE 4 EXAMPLE OF SHIPBOARD DATA INVENTORY/TRANSFER LOG

SECTION TWO

DATA LOGGING AND ANNOTATING REQUIREMENTS AND PROCEDURES
(OUTLINE)

I. ORIENTATION FOR WATCHSTANDERS

- A. Two main tasks of watchstanders
- B. Types of logsheets used

II. GENERAL DEFINITIONS AND POLICIES FOR DATA RECORDING

- A. Use of Julian date and GMT
- B. Definition of Cruise Locator and area code
- C. Definition of Archive No. and Sample No.
- D. Definition and numbering of stations and samples collected
- E. Definition and numbering of tracklines
- F. Definition of "required" and "desirable" logbook entries
- G. Definition of the log entries that will be keypunched

III. ENTRIES INTO LOGBOOKS

- A. Navigation Log
- B. Acoustic Equipment Log
- C. Geopotential Sensor Log
- D. Station Operations Log
- E. Cruise Data Index Log

IV. ANNOTATION OF ANALOG RECORDS

- A. General rules for all records

V. LABELLING OF SAMPLES

- A. Affixing "Sample No." to sample container
- B. "Required" and "desired" information on sample labels

VI. LABELS FOR ALL PHYSICAL DATA

- A. General rules for use of labels
- B. Analog record boxes/bags
- C. Magnetic tape reels
- D. Magnetic tape boxes
- E. Physical station samples
- F. All other physical data

DATA LOGGING AND ANNOTATING REQUIREMENTS AND PROCEDURES

I. ORIENTATION FOR WATCHSTANDERS

A. Watchstanders have two main tasks:

1. Logging - Where information about data or samples being collected is written on logsheets which pertain to particular data systems or to sample operations. Resulting logbooks enable an investigator to get pertinent information about data or samples without unrolling a long paper roll or without actually getting the physical samples out of the sample archives.
2. Annotating records - labeling, stamping or writing information pertinent to data interpretation and cataloguing directly on to a data record which is usually an analog (continuous trace data) paper roll. This helps an investigator orient himself when working directly with a record.

B. There are 5 main types of logsheets used on Branch of Marine Geology cruises out of Menlo Park, CA.

1. Navigation - Filled in by people assigned as navigation watchstanders. Required unless contract navigation is used.
2. Acoustic Equipment - These are used for all seismic systems including sparker, airgun, fathometer, uniboom, burg, minisparker, etc. Use one set of logsheets for each system operated on your cruise. They should be filled out by the watchstander assigned to a particular data system.
3. Geopotential Sensor - These are used to monitor magnetometer, gradiometer or gravimeter data systems. They may also be used for multi-instrument systems which are recording bathymetry and navigation along with magnetics and gravity on a single recording medium.
4. Station Operations - Used to keep track of all sampling and data gathering activities while 'on station'. Both the sample or data type and when the sample or data was collected is entered in this log. Samples may be rocks, sediment, or water. This logsheet may also be used to record the deployment of sonobuoys, on-the-bottom seismometers, explosives,

geoprobes, etc., as well as the operation of television or cameras and the recording of various types of hydrographic numerical information.

5. Cruise Data Index Log - This log is used by the navigator to record start/end times for tracklines (also entered in navigation log) as well as recording times for various kinds of navigation data. It is also used by the Cruise Curator to enter personnel and equipment lists, etc.

II. GENERAL DEFINITIONS AND POLICIES FOR DATA RECORDING

A. Use the Julian date and Greenwich Mean Time (GMT) for all day/time entries in logs or on records. The Julian day is the count of the day of the year taking January 1st as day one and December 31st as day 365. Julian day changes are keyed to GMT, not to local time.

B. The "Cruise Locator" is the number assigned prior to the beginning of each cruise that uniquely identifies the cruise. The format is: ship (normally designated by 1st letter of the ship's name); cruise no. for the ship in a given year; -year (e.g., S4-76; the fourth cruise of the SOUNDER in 1976). The "Cruise Locator" _must appear_ on all data (logsheets, samples, maps, rolls, etc.) collected or processed during the cruise, to ensure that the data is correctly identified with the cruise it was collected on. The "area" code is a two letter code that designates the general operational area of the cruise. The codes are based on Branch of Marine Geology project areas. Although the use of the code is optional, space for it is provided on all forms and labels to provide data users with a quick reference as to where the data were collected. The codes are (see figure 1):

MX - Mexico
SC - southern California
NC - northern California
WO - Washington/Oregon
NP - northern Pacific
EG - eastern Gulf of Alaska
WG - western Gulf of Alaska
BS - Bering Sea
AR - Arctic: Chukchi and Beaufort Seas
HW - Hawaiian Islands

C. The "Archive No." is a unique letter/number combination assigned to each different type of physical sample or recorded data collected during a cruise. It has two functions: 1. It is the basis for all post-cruise hierarchal data searches, inventory listings, etc. 2. It is the number in combination with the Cruise Locator that all data, except physical samples, will be stored under in the data archives. The format of the "Archive No." is:

Data Code - Sequence No.

The Sequence Number is the roll, reel, or sample attempt, etc., number. (e.g., UNIB - 14; or SCAG - S10)

"Sample No.'s" are used to uniqely identify physical samples stored in U.S.G.S. sample archives. The format of the "Sample No." is:

Cruise Locator - Sampling Attempt No.**

(e.g., S4 - 76 - 103A; or S4 - 76 - D10; etc.)

**See item D below for further details on sample numbering.

D. A "station" (geologic, geophysical, biologic, land, etc.) shall be defined as one or more sampling attempts of the ocean bottom, water column, or land outcrop at a specific location. Each "sampling attempt" is associated with a unique time; consequently, it is essential that each "sampling attempt" at a "station" have a unique "sequence number" (any combination of letters and/or numbers up to five) regardless of whether a sample is recovered.

It is preferable that "sampling attempts" be numbered consecutively from the beginning of each new cruise. Two numbering systems are proposed although other systems may be used if and only if they assign a unique "sequence" number to each "sampling attempt":

SYSTEM A:

1. When only one "sampling attempt" is made at a "station", the "station" number is synonomous with the "sampling attempt" number. (e.g., "stations" = 3, 4, 5.....; "sampling attempts" = 3, 4, 5.....)
2. When multiple "sampling attempts" are made at a "station", the "station" number is synonomous with the first "sampling attempt" number at that station. (e.g., "station" = 3; "sampling attempts = 3, 3A, 3B.....)

SYSTEM B:

1. Regardless of the number of "sampling attempts" at each "station", the "sampling attempts" and "stations" are numbered separately. (e.g., "station = 3; "sampling attempts" = D2, P1, C3.....)

Log entries for the start/end times of stations are optional for all logs except the navigation log, regardless of which sample numbering system is used. However, station numbers must be entered in the "STA. NO./SHOT POINT" field (columns 16-19) of the Station Operations log for each sample collected on a given station. Start/end times for stations may be entered in either the Station Operations log or the Cruise Data Index log in the format:

Spec. Code - Data Code - Sta. No
(e.g., NAV - STAT - 3)

Samples recovered under either system A or B sould be numbered according to the following format:

1. On the Station Operations Log:
   Spec./Data Code - "sampling attempt" no.
2. On the sample label or sample bag:
   Cruise Locator - "sampling attempt" no.

This is the "Sample No." as defined in II.C. on page 19 which uniquely identifies each sample collected on a cruise. Examples of sample numbering for a dart core from Cruise "S4-76" are:

	Station Operation Log	Sample Container*
SYSTEM A:	SAM - DART - 3A	S4 - 76 - 3A
SYSTEM B:	SAM - DART - D2	S4 - 76 - D2

*For complete description of sample labelling, refer to Part V. of this section (page 38).

E. A trackline shall be defined as continuous ship movement along a predetermined track (preferably of a single bearing) for the purpose of data recovery.

1. All tracklines must have a unique number and it is preferable that tracklines be numbered consecutively as they are run. However, other numbering systems may be used as long as each number is unique.

2. Trackline numbers should change whenever:
    a. A predetermined trackline is completed.
    b. A predetermined trackline is disrupted before completion. For instance, if the ship does not complete a given line due to cross traffic, equipment failure, etc., then a new number should be assigned to the remainder of the disrupted line; the new number may be composed of the old line number plus a letter.
3. Start and end of tracklines should be logged in the U.S.G.S. Cruise Data Index log by the navigator.

F. Definition of "required" and "desirable" logbook entries.
1. "Required" logbook entries include information essential to the Marine Data Information system and necessary for the correct interpretation of records. These entries must be made by the watchstander as soon as possible after the logging event.
2. "Desirable" logbook entries include information useful to the Chief Scientist in both data interpretation and in understanding ship operations. These entries shall be required if the Chief Scientist feels it is necessary to enter them in a particular log.

G. Definition of the logbook entries to be keypunched: Certain logbook entries will be keypunched directly from field logbooks in order to produce an accurate computerized index and inventory of data collected on Branch of Marine Geology cruises. This index will also be used to plot data locations and to generate statistical cruise reports. Keypunchable entries include:
1. All entries in the Cruise Data Index Log.
2. Entries in the Acoustic and Geopotential Equipment, and Station Operations Logs that have a mark in the "Punch Flag" box.
    a. Start/end times for all paper rolls, magnetic tapes, printer listings, etc.
    b. On/off times for all periods when data is being recorded on items in "a". This includes on/off times (using "SYS" codes) for individual data system components that are

being recorded with other data systems on the same recording medium.

c. In Station Operations Log for time(s) that actual sample attempts or numerical observations are made.

III. ENTRIES INTO LOGBOOKS (Detailed examples of how to fill out individual logsheets are on pages 24-26, 29, 31, 34, and 41)

A. Navigation Logbook (see page 25 for example)

1. General rules
   - a. Logbook optional when contract navigation is being used.
   - b. Use one logbook per cruise
   - c. All entries must be sequential by time.
2. Logbook entries required at:
   - a. Times for all navigation fixes
   - b. All CSE and SPD changes while on trackline
   - c. Start/end times of all stations
   - d. Start/end times of all tracklines*
   - e. Start/end times of navigation equipment failures*
3. Specific procedures: For a more detailed explanation of some of the considerations in monitoring navigation equipment and logs (as well as gravity/magnetics and bathymetry), see the section: SPECIAL CONSIDERATIONS FOR NAVIGATION/GRAVITY/MAGNETICS/BATHYMETRY WATCHES on page 42.

*These items must also be recorded in the CDI log.

B. Acoustic Equipment Log (sparker, airgun, fathometer, etc.)

1. General rules (see page 26 for example)
   - a. One set of logsheets for each acoustic instrument system
   - b. A routine entry which includes all logsheet parameters plus the name of the new watchstander should be made at the start of each 4 hour watch - more frequent routine entries at discretion of Chief Scientist.
2. Additional "required" logsheet entries (see page 21)
   - a. Cruise Locator
   - b. Start/end times of: (*for these entries, the "punch flag" box must be marked)
     - *1) Analog rolls and printer listings
     - *2) Magnetic and paper tapes
     - *3) Periods when data is being recorded on items 1) and 2) above. Be sure to enter the proper "Media/Spec."

FIGURE 5 EXAMPLE OF HOW TO FILL OUT PORTIONS OF ALL LOGS TO BE KEYPUNCHED

DATA FIELDS COMMON TO ALL LOGS (WITH MINOR EXCEPTIONS)

DATA FIELDS COMMON TO CDI AND STATION OPERATIONS LOGS

CRUISE LOCATOR L1 - 77 - NC (COL 73-80)
10 YR AREA

LOGS ENTRIES REQUIRED AT:	TIME & POSITION				*	MEDIA/ SPEC CODE	ARCHIVE NO		'X' IN ONE BOX ONLY					DATA DESCRIPTION	STATION DATA ONLY		
	JULIAN	GMT TIME		LINE NO.	STA. NO		DATA/EQUIP CODE	ROLL, REEL OR SEQUENCE NO	UNIQ SMPL	S/E ROLL, SMPL, ETC		DATA/SYSTM RECORDING			RECOVERY?	WATER DEPTH	DEPTH UNIT
	DAY	HR-MIN	SEC.		SHOT POINT	3 LET	4 LETTERS	5 LET/NO *	TIME	STRT	END	ON	OFF		Y/N	4 NO *	LET
a. START/END TIMES OF RECORDING ON ALL ANALOG RECORDS	100	1137		12		PAP	UNIB	13		X							
	100	1643				PAP	UNIB	13			X						
b. START/END TIMES OF RECORDING ON ALL MAGNETIC TAPES	100	1132				MAG	MC24	4		X							
	100	1751				MAG	MC24	4			X						
c. ON/OFF TIMES OF RECORDING INTERVALS WITHIN ANALOG RECORDS OR MAGNETIC TAPES	100	1309				PAP	UNIB	13					X				
	100	1315				PAP	UNIB	13				X					
d. START/END TIMES OF RECORDING ON TAPES OR ROLLS FOR MULTI-INSTRUMENT DATA	100	0832				MAG	GRV2	3		X							
	100	1458				MAG	GRV2	3			X						
e. ON/OFF TIMES OF SINGLE COMPONENTS OF A MULTI-INSTRUMENT SYSTEM	100	1200				SYS	BA35						X				
	100	1219				SYS	BA35					X					
f. START/END TIMES OF TRACKLINES	100	0639				NAV	TRAK	12		X				START TRACKLINE 12			
	100	1823		↓		NAV	TRAK	12			X			END TRACKLINE 12			
g. START/END TIMES OF SAMPLES COLLECTED OVER A TIME INTERVAL	102	2313		15	250	SAM	CHAN	250D1		X				START DREDGE HAUL		150	M
	102	2345			250	SAM	CHAN	250D1			X			120 #S PHOSPHRITE NODULS	Y	80	M
h. THE TIME OF A DISCRETE SAMPLING EVENT (UNIQUE TIME)	103	0051		↓	251	SAM	DART	251C1	X					BENT BARREL 15cm SH CHIP	Y	308	M
	1-3	5-8	9, 10	12-15	16-19	21-23	25-28	29-33	36	37	38	39	40	41-64	66	67-70	71

* PLEASE RIGHT - JUSTIFY THESE ENTRIES

GENERAL RULES

1. Entries may be made in any order or at any time.

2. Numbers must be right justified (e.g., put numbers at far right side of each data field).

3. Aids to keypunching:

   a. Use block letters and make numbers as clear as possible.
   b. A column to be keypunched must be filled with a number/letter value or have a clear-cut continuation arrow from an earlier entry. (see 'LINE NO.' field)
   c. Letter 'O' is entered as 'Ø'
   d. The letter 'I' should be clearly distinguished from the number '1'
   e. Do not use dittos. They might look like an "eleven" to the keypunch operator.

4. Use pencil to fill in logs so corrections can be made.

DATA FIELD DESCRIPTIONS AND RULES

TIME & POSITION DATA:
1. Julian Day changes are keyed to GMT not local time.
2. GMT time goes from '0000 00' to '2359 59'
3. 'LINE / STA.NO.-SHOT POINT' fields are used to indicate what trackline, station or shot point a logged event takes place at. This field on the 'Station Operations Log' must be filled out.

MEDIA/SPEC. CODE: A three letter mnemonic code that designates what type of recording media (e.g., paper rolls, mag tape, etc.) or misc. specification (e.g., navigation data, personnel lists, etc.) is being logged. See 'Cruise Curators Manual' for complete list of codes.

ARCHIVE NUMBER: This is the number used for both the computer organization of cruise data and for locating cruise data in the permanent archives - please use care in entering this information.
1. DATA/EQUIP CODE: A four letter mnemonic code that designates the data type (e.g., tracklines, samples, etc.) or equipment (e.g., uniboom, dart core, etc.) that is being logged. See 'Cruise Curators Manual' for complete list of codes.
2. SEQUENCE NO.: The number assigned to each type of data (e.g., roll no., line no., sample no., etc.)
   a. This number must be unique for each piece of dsta of s given dats type. Normal convention is to use consecutive numbers from beginning of cruise.
   b. May be combination of numbers and letters if necessary.

'X' IN APPROPRIATE BOX: Check ONE box only, relating to the information being recorded.
1. Whenever the 'S/E Roll, Sample' column is checked, it is assumed that the Data/System has just started or ended its recording on a given roll, reel, etc., or that a sample (i.e., dredge) has just started or stopped collecting a sample.
2. A check in the 'ON/OFF' columns in the 'Data/System recording field, indicates that data or system has stopped or started recording within a roll, reel, etc.

DATA DESCRIPTION: Generally a free field description of the data being entered. (Only on CDI and Station Operations Logs)
1. Cruise/Port stops
2. Inventory lists of personnel and scientific equipment.
3. Instrument readings and sample information

STATION DATA ONLY: Make entries in these columns only when station data (e.g., cores, grabs, sonobuoys, etc.) are being recorded.
1. RECOVERY: A 'Y' (yes) means that data was collected (recovered).
2. DEPTH UNIT: The code letters for this column are:
   M = Meters
   F = Fathoms
   X = Feet
   S = Seconds (two-way time)

DATA FIELD DESCRIPTIONS

1. **Fix Quality**

G = Good		
F = Fair		
Q = Questionable		Fix type "DR" must be either "P" or "Q"
P = Poor		Fix type "DRC" must be "P", "Q", or "F"

2. **Direction Code**

A Flag used to indicate large course changes (more than 10°) or large speed changes (more than 1 kt). This information is used to smooth the navigation for tightly controlled surveys.
   a. Use "1" (one) Start /End Line or large CSE/SPD change
   b. Otherwise leave blank

3. **Fix Type** (cols. 41-43) Left justify

A - Celestial	LCR - Loran C R-R
R - Radar	DR - Dead Reckoning
S - Satellite	DRC - Dead Reckoning with other positioning control
V - Visual	CC - Change course
CØ - Contractual*	CS - Change speed
LA - Loran A	CCS - Change course and speed
LC - Loran C	SØ - Steady on course/speed

* When contractural Fixes are logged, columns 32 thru 35 in the "comments" section are reserved for the contractors Fix number.

4. **Speed** (cols. 49-52)

   a. speed in knots
   b. right justify

5. **Latitude and Longitude**

   a. recorded in degrees, minutes and hundreths of minutes
   b. conventions: Latitude + North - South
      Longitude + East - West

6. **Wind**

   a. "Dir" is in asimuthal degrees (0-360°; relative to true north)
   b. "SPD" is in knots
   c. Dir/Spd are true Dir/Spd (e.g. measured values corrected for ships heading and speed)

7. **Visual, Radar, Celestial**

   a. "Angles" is the measured angle between two objects
   b. "Brgs" is the asimuthal direction to an object (0-360°; relative to true north)
   c. "Distance" is the distance of the object from the ship (in yards)

8. **Loran**

   a. "Rate" is the station channel number used for the Loran Reading
   b. "Reading" is a four digit number; read directly from the Loran receiver
   c. Ts or Tg: indicate whether reading is from "Sky" (Ts) or "ground" (Tg) waves
   d. Time +/-min: indicates how much time elapsed between readings made on different channels
   e. GFP: Good/Fair/Poor - indicates the quality of each channel reading

GENERAL RULES FOR FILLING OUT NAVIGATION LOG:

1. All entries must be sequential by time
2. Logbook entries required at
   a. All CSE and SPD changes while on tracklines
   b. Start/End times of all tracklines (also recorded in CDI log)
   c. Start/End times of all stations
   d. Times for all SAT-NAV fixes
3. Keypunch conventions
   a. Letter "O" is entered as Ø
   b. Letter "I" is entered as I
   c. Number "1" is entered as 1

U.S.G.S. NAVIGATION LOG

DATE 8/78

Cruise Locator L1 76 MX
ID. YR AREA

Ship R/V S.P. Lee Chief Scientist Bob Rowland Affiliation U.S.G.S.

Page 5 of 31

Julian Day	GMT TIME HR	MIN	SEC	LINE NO	STATION NO	COMMENTS	Fix Quality	Dir. Code	Fix Type	New Course	New Speed	LATITUDE DEG	LATITUDE MINUTES	LONGITUDE DEG	LONGITUDE MINUTES	WIND Dir.	WIND SPD	VISUAL, RADAR, ETC. OBJECT	Angles	Brgs	Distance	LORAN, RAYDIST, etc Rate	Reading	TS/TG	Time +/- min	GFP
300	10	15	00	3		ST LINE	F	1	V	295	8.5	38	2135	-142	1010	210	17	Pt Goorl		136	2050					
300	10	48	30	3		SAT FIX	G		S	295		38	2315	-142	1210			Blk Rck		185	2470					
	11	01	00	3	11	UNDERWAY SAMPLE	Q		DR	295		38	2411	-142	1305			Mr Smp		C37	4070					
	11	05	40			AVOID TRAFFIC		1	CC	280	8.5															
	11	07				MANEUVER		1	CCS	275	5.0															
	11	12				RESUME LINE		1	CCS	245	8.5															
	11	13	30			ON LINE		1	SØ	245	8.5															
	13	05	40			SAT FIX			S	245	8.5	38	2701	-142	1595											
	14	05				END LINE	Q	1	DRC	245	8.5	38	3035	-142	1708											
	14	05		3		TURN TO LNE 4			CCS	095	8.5															
	14	10		4		ON LINE		1	SØ	095	8.5															
	14	10				CNTRCT FIX 994	G	1	CØ																	
	14	15				CNTRCT FX1000	G		CØ																	
300	14	20		4		LOR C FIX	F		LC	095	8.5	38	4215	-142	1793							1H4	6341	Tg	0	G
																						2H3	1495	Ts	+15	P

FIGURE 6 EXAMPLE OF HOW TO FILL OUT NAVIGATION LOG (see page 23)

GENERAL RULES FOR FILLING OUT THE ACOUSTIC EQUIPMENT LOG:

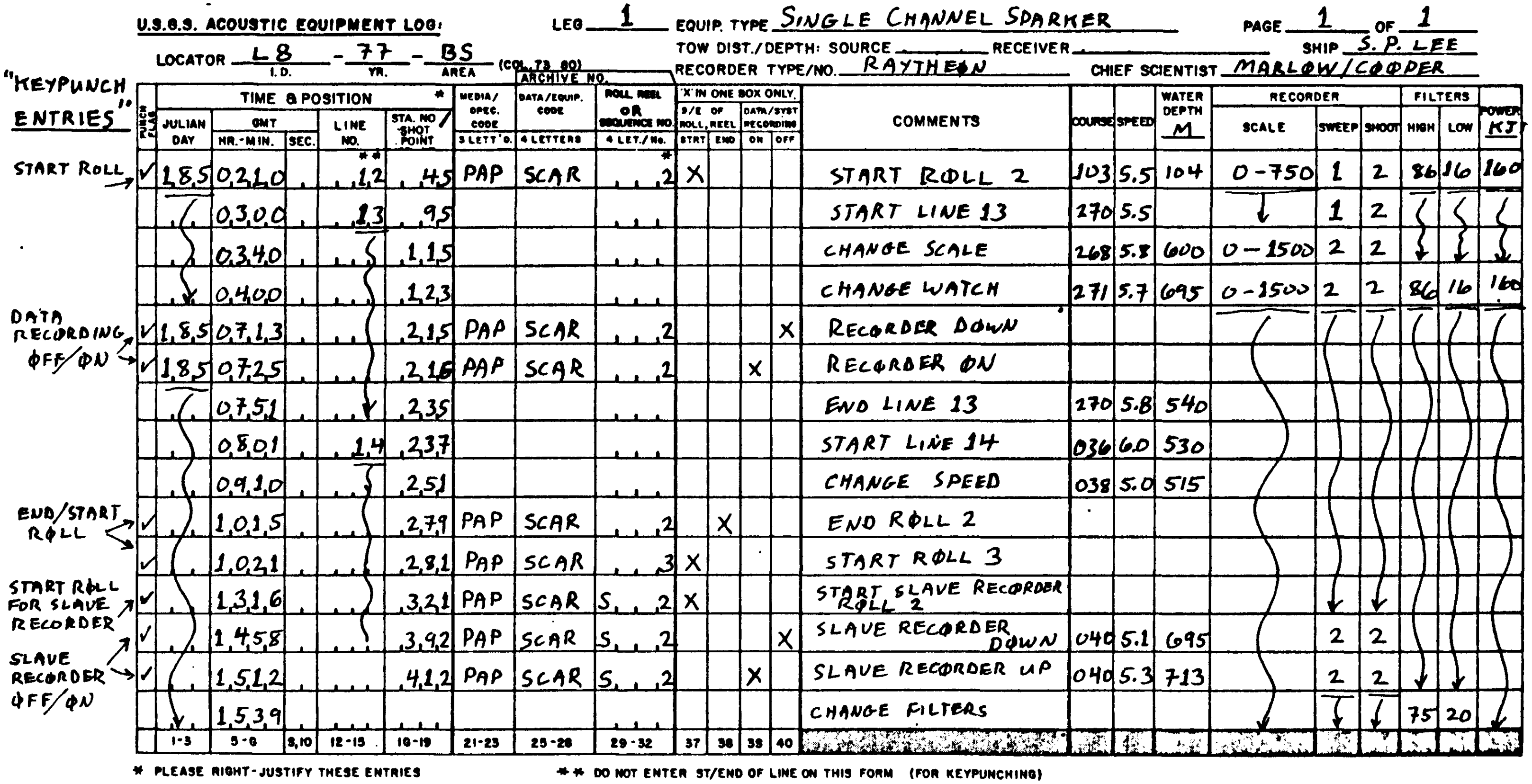
1. GENERAL RULES AND DATA FIELD DESCRIPTIONS LISTED ON SHEET DESCRIBING HOW TO FILL OUT PORTIONS OF LOGS TO BE KEY-PUNCHED, ALSO APPLY TO THIS LOG. (see page 24)

2. ENTRIES REQUIRED AT:
   - **A. START/END TIMES FOR ALL PAPER ROLLS, MAG TAPES, PRINTER LISTS, ETC.
   - **B. ON/OFF TIMES FOR ALL PERIODS DURING WHICH:
     1. DATA IS BEING RECORDED ON ITEMS LISTED IN A.
     2. COMPONENTS OF MULTI-INSTRUMENT SYSTEM OPERATE.
   - C. OTHER SPECIFIC TIMES INCLUDING:
     1. THE BEGINNING OF EACH 4 HOUR WATCH PERIOD.
     2. AN EVENT AFFECTING THE RECORDING INSTRUMENT STATUS.
     3. WHENEVER SPECIFIED BY THE CHIEF SCIENTIST.

3. FOR ENTRIES WHERE THE 'PUNCH FLAG' BOX IS MARKED, ALL COLUMNS ON THE LEFT SIDE OF THE LOG SHEET (COL. 1-40) SHOULD BE WRITTEN OUT; DO NOT USE DITTOS. THEY CONFUSE KEY-PUNCHER.

4. FOR A SLAVE RECORD, PUT AN 'S' AT THE FAR LEFT SIDE OF THE 'ROLL/REEL NO.' COLUMN.

** FOR THESE ITEMS, AND ONLY THESE ITEMS, AN 'X' OR '✓' SHOULD BE ENTERED INTO THE 'PUNCH FLAG' COLUMN (FAR LEFT COLUMN)



U.S.G.S. ACOUSTIC EQUIPMENT LOG: LEG 1 EQUIP. TYPE SINGLE CHANNEL SPARKER PAGE 1 OF 1

LOCATOR L8 (I.D.) - 77 (YR.) - BS (AREA) (COL. 73-80) TOW DIST./DEPTH: SOURCE ____ RECEIVER ____ SHIP S. P. LEE

RECORDER TYPE/NO. RAYTHEØN CHIEF SCIENTIST MARLØW/CØØPER

"KEYPUNCH ENTRIES"	PUNCH FLAG	TIME & POSITION *					MEDIA/OPEC. CODE	ARCHIVE NO. DATA/EQUIP. CODE	ROLL REEL OR SEQUENCE NO.	'X' IN ONE BOX ONLY.				COMMENTS	COURSE	SPEED	WATER DEPTH M	RECORDER			FILTERS		POWER KJ
		JULIAN DAY	GMT		LINE NO.	STA. NO/SHOT POINT				S/E OF ROLL, REEL		DATA/SYST RECORDING						SCALE	SWEEP	SHOOT	HIGH	LOW	
			HR.-MIN.	SEC.	**		3 LETT'S.	4 LETTERS	4 LET./No. *	STRT	END	ON	OFF										
START ROLL	✓	185	0210		12	45	PAP	SCAR	2	X				START RØLL 2	103	5.5	104	0-750	1	2	86	16	160
			0300		13	95								START LINE 13	270	5.5			1	2			
			0340			115								CHANGE SCALE	268	5.8	600	0-1500	2	2			
			0400			123								CHANGE WATCH	271	5.7	695	0-1500	2	2	86	16	160
DATA RECORDING ØFF/ØN	✓	185	0713			215	PAP	SCAR	2				X	RECØRDER DØWN									
	✓	185	0725			216	PAP	SCAR	2			X		RECØRDER ØN									
			0751			235								END LINE 13	270	5.8	540						
			0801		14	237								START LINE 14	036	6.0	530						
			0910			251								CHANGE SPEED	038	5.0	515						
END/START RØLL	✓		1015			279	PAP	SCAR	2		X			END RØLL 2									
	✓		1021			281	PAP	SCAR	3	X				START RØLL 3									
START RØLL FOR SLAVE RECORDER	✓		1316			321	PAP	SCAR	S 2	X				START SLAVE RECØRDER RØLL 2									
SLAVE RECØRDER ØFF/ØN	✓		1458			392	PAP	SCAR	S 2				X	SLAVE RECØRDER DØWN	040	5.1	695		2	2			
	✓		1512			412	PAP	SCAR	S 2			X		SLAVE RECØRDER UP	040	5.3	713		2	2			
			1539											CHANGE FILTERS							75	20	
		1-3	5-6	8,10	12-15	16-19	21-23	25-28	29-32	37	38	39	40										

* PLEASE RIGHT-JUSTIFY THESE ENTRIES

** DO NOT ENTER ST/END OF LINE ON THIS FORM (FOR KEYPUNCHING)

FIGURE 7 EXAMPLE OF HOW TO FILL OUT ACOUSTIC EQUIPMENT LOG (see page 23)

and "Data" codes used to identify the data type being logged (see pages 61-68).

*4) Periods when equipment for a multi-instrument system is in operation (e.g., if both uniboom and bathymetry data are being recorded simultaneously on the same record and the uniboom becomes inoperative, then the fact that the uniboom went off must be recorded. "Spec." code used = 'SYS')

c. Change in recorder (do not mark "punch flag" box)

1) Scale

2) Sweep rate

3) Shoot rate

d. Change in equipment status (do not mark "punch flag" box)

1) Filter settings

2) Power available

3) Source depth

4) Receiver depth

3. "Desirable" logsheet entries - to be entered at the discretion of the Chief Scientist: (do not mark "punch flag" box)

a. Start/end times of all tracklines

b. Change in ship's course or speed

C. <u>Geopotential Sensor Log</u> (see page 29 for example)

1. General rules

a. More than one type of geopotential instrument (gravimeter, gradiometer, magnetometer, etc.) may be entered on the same set of logsheets

b. A routine entry which includes all logbook parameters plus the name of the new watchstander should be made at the start of each 4-hour watch; more frequent routine entries at descretion of Chief Scientist.

c. For specific considerations in monitoring geopotential logs and equipment see: SPECIAL CONSIDERATIONS FOR NAVIGATION/GRAVITY/MAGNETICS/BATHYMETRY WATCHES on page 42.

2. Additional "required" logbook entries (see page 21)

a. Cruise Locator

b. Start/end times of (*for these entries, the "punch flag" box must be marked)

*1) Analog rolls and printer listings

*2) Magnetic and paper tapes

*3) Periods when data is being recorded on items 1) and 2) above. Be sure to enter the proper "Media/Spec." and "Data" codes used to identify the data type being logged (see pages 61-68)

*4) Periods when equipment for a multi-instrument system is in operation (e.g., if bathymetry and gravity are being recorded on the same magnetic tape and the bathymetry goes off, then the fact that bathymetry is no longer being recorded must be logged, "Spec. Code" will = 'SYS'). Be sure to use proper "Media/Spec." and "Data" codes.

c. Change in equipment status (do not mark "punch flag" box)

1) Number of sensors operating

2) Sensor towing distance

3. "Desirable" logbook entries - to be entered at discretion of Chief Scientist: (do not mark "punch flag" box)

a. Start/end times of all tracklines

b. Change in ship's course or speed

GENERAL RULES FOR FILLING OUT THE GEOPOTENTIAL LOG:

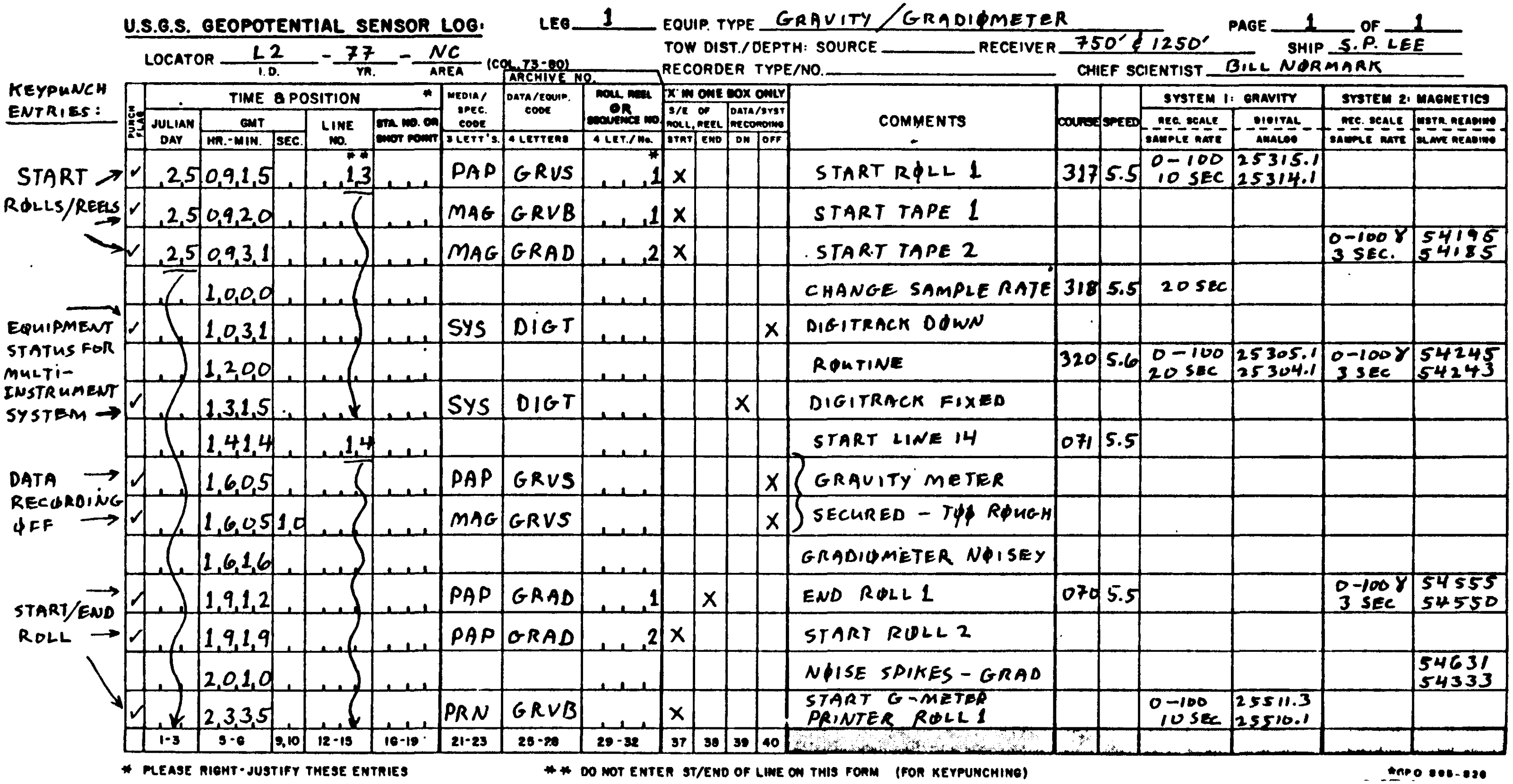
1. GENERAL RULES AND DATA FIELD DESCRIPTIONS LISTED ON SHEET DESCRIBING HOW TO FILL OUT PORTIONS OF LOGS TO BE KEY-PUNCHED, ALSO APPLY TO THIS LOG. (see page 24)

2. ENTRIES REQUIRED AT:
   - **A. START/END TIMES FOR ALL PAPER ROLLS, MAG TAPES, PRINTER LISTS, ETC.
   - **B. ON/OFF TIMES FOR ALL PERIODS DURING WHICH:
     1. DATA IS BEING RECORDED ON ITEMS LISTED IN A.
     2. COMPONENTS OF MULTI-INSTRUMENT SYSTEM OPERATE.
   - C. OTHER SPECIFIC TIMES INCLUDING:
     1. THE BEGINNING OF EACH 4 HOUR WATCH PERIOD.
     2. AN EVENT AFFECTING THE RECORDING INSTRUMENT STATUS.
     3. WHENEVER SPECIFIED BY THE CHIEF SCIENTIST.

3. FOR ENTRIES WHERE THE 'PUNCH FLAG' BOX IS MARKED, ALL COLUMNS ON THE LEFT SIDE OF THE LOG SHEET SHOULD BE WRITTEN OUT; DO NOT USE DITTOS.

4. FOR A SLAVE RECORD, PUT AN 'S' AT THE FAR LEFT SIDE OF THE 'ROLL/REEL NO.' COLUMN.

** FOR THESE ITEMS, AND ONLY THESE ITEMS, AN 'X' OR '✓' SHOULD BE ENTERED INTO THE 'PUNCH FLAG' COLUMN (FAR LEFT COLUMN)



U.S.G.S. GEOPOTENTIAL SENSOR LOG: LEG 1 EQUIP. TYPE GRAVITY /GRADIØMETER PAGE 1 OF 1

LOCATOR L2 - 77 - NC (COL. 73-80)
I.D. YR. AREA

TOW DIST./DEPTH: SOURCE ______ RECEIVER 750' & 1250' SHIP S.P. LEE

RECORDER TYPE/NO. ______ CHIEF SCIENTIST BILL NØRMARK

KEYPUNCH ENTRIES:	PUNCH FLAG	TIME & POSITION *					MEDIA/ SPEC. CODE	ARCHIVE NO.		'X' IN ONE BOX ONLY				COMMENTS	COURSE	SPEED	SYSTEM 1: GRAVITY		SYSTEM 2: MAGNETICS	
		JULIAN DAY	GMT		LINE NO. **	STA. NO. OR SHOT POINT		DATA/EQUIP. CODE	ROLL, REEL OR SEQUENCE NO. *	S/E OF ROLL, REEL		DATA/SYST RECORDING					REC. SCALE SAMPLE RATE	DIGITAL ANALOG	REC. SCALE SAMPLE RATE	MSTR. READING SLAVE READING
			HR.-MIN.	SEC.			3 LETT'S.	4 LETTERS	4 LET./No.	STRT	END	ON	OFF							
START →	✓	25	0915		13		PAP	GRVS	1	X				START RØLL 1	317	5.5	0-100 10 SEC	25315.1 25314.1		
RØLLS/REELS →	✓	25	0920				MAG	GRVB	1	X				START TAPE 1						
	✓	25	0931				MAG	GRAD	2	X				START TAPE 2					0-100 γ 3 SEC.	54196 54185
			1000											CHANGE SAMPLE RATE	318	5.5	20 SEC			
EQUIPMENT STATUS FOR MULTI-INSTRUMENT SYSTEM →	✓		1031				SYS	DIGT					X	DIGITRACK DØWN						
			1200											RØUTINE	320	5.6	0-100 20 SEC	25305.1 25304.1	0-100 γ 3 SEC	54245 54243
	✓		1315				SYS	DIGT				X		DIGITRACK FIXED						
			1414		14									START LINE 14	071	5.5				
DATA RECORDING ØFF →	✓		1605				PAP	GRVS					X	GRAVITY METER						
→	✓		1605	10			MAG	GRVS					X	SECURED - TØØ RØUGH						
			1616											GRADIØMETER NØISEY						
START/END ROLL →	✓		1912				PAP	GRAD	1		X			END RØLL 1	070	5.5			0-100 γ 3 SEC	54555 54550
→	✓		1919				PAP	GRAD	2	X				START RØLL 2						
			2010											NØISE SPIKES - GRAD						54631 54333
	✓		2335				PRN	GRVB		X				START G-METER PRINTER RØLL 1			0-100 10 SEC	25511.3 25510.1		
		1-3	5-6	9,10	12-15	16-19	21-23	25-28	29-32	37	38	39	40							

* PLEASE RIGHT-JUSTIFY THESE ENTRIES

** DO NOT ENTER ST/END OF LINE ON THIS FORM (FOR KEYPUNCHING)

☆GPO 808-820

FIGURE 8 EXAMPLE OF HOW TO FILL OUT GEOPOTENTIAL LOG (see page 27)

D. Station Operations Log (see 31 for example)

1. General Rules

   a. A single 'Station Operations log' will be used to keep track of all data gathering or sample collecting activities on a 'station'.

      Pertinent log entries indicate the type of data or samples collected and the time or the place they are collected (stations may be geologic, hydrologic, biologic, television, camera, or geophysical).

   b. All sampling or data gathering activities for a single 'station' shall be recorded on only one sheet of the Station Operations log (except where continuation sheets are necessary).

2. "Required" logbook entries

   a. Cruise Locator

   b. "Station No." (in columns 16-19, headed by 'STA. NO. or SHOT POINT')

   c. Time interval over which a sample is actually being collected or data gathered - only one entry for a discrete sampling event or data observation, as when a dart core hits bottom; but two entries bracketing the time interval that a dredge is dragging over bottom. Also include the associated information such as:

      1) 4-letter Data Code/Equipment Code and the 3-letter Media/Spec. Code (see pages 61-68)

      2) "Sampling Attempt" number (see pages 19 & 20)

      3) Indication of sample recovery

      4) Water depth

      5) An "X" in the "discrete" sampling field (column 36) for a "discrete sampling event".

      6) An "X" in either the "start" or "end" Sampling Location field (columns 37 or 38) to signal the start or end of a dredge haul.

GENERAL RULES FOR FILLING OUT THE STATION OPERATIONS LOG:

1. GENERAL RULES AND DATA FIELD DESCRIPTIONS LISTED ON SHEET DESCRIBING HOW TO FILL OUT PORTIONS OF LOGS TO BE KEY-PUNCHED, ALSO APPLY TO THIS LOG. (see page 24)
2. THIS LOG IS TO BE USED TO MONITOR GEOLOGIC, HYDROLOGIC, BIOLOGIC, TELEVISION, CAMERA, OR GEOPHYSICAL (I.E., SONOBUOYS) STATION OPERATIONS.
3. CHECK 'PUNCH FLAG' BOX ONLY FOR TIME(S) THAT ACTUAL SAMPLE ATTEMPTS OR NUMERICAL OBSERVATIONS ARE MADE.
4. USE ONLY ONE LOGSHEET PER STATION UNLESS MORE SPACE IS NEEDED.
5. THE STATION # MUST BE ENTERED IN THE 'STA.NO.' FIELD FOR EVERY ENTRY WITH A '✓' IN THE 'PUNCH FLAG' BOX.
6. LOGGING 'START/END' OF STATIONS IS OPTIONAL.
7. NUMERICAL OBSERVATION DATA (SEE EXAMPLE BELOW) MAY BE ENTERED IN 'OPERATIONS, COMMENTS, OR DATA MEASUREMENTS' FIELD. THE 'SPEC CODE' FOR THESE ENTRIES IS 'OBS'.

U.S.G.S. STATION OPERATIONS LOG

STATION # 57

SHIP/LEG SEA SØUNDER/II AREA NØRTØN SØUND

LAT (ST/ED) 63° 29' 10"/63° 29' 12"

CHIEF SCI. HANS NELSØN CRUISE LOCATOR (COL 73-78) 54 (ID) - 77 (YR) AREA CODE (COL. 79-80) BS

LONG (ST/ED) -165° 48' 31"/-165° 48' 33"

KEYPUNCH ENTRIES:	PUNCH FLAG	TIME & POSITION *				MEDIA/SPEC CODE	DATA/EQUIP CODE	SEQUENCE/SAMPLE ** ATTEMPT NO	'X' IN ONE BOX ONLY SAMPLING LOCATION			OPERATIONS, COMMENTS, OR DATA MEASUREMENTS	STATION DATA ONLY		
		JULIAN DAY	GMT TIME HR-MIN	LINE NO	STA NO OR SHOT POINT				DISCRETE E.G. CORE	VARIABLE E.G. DREDGE STRT	END	INSTRUMENT DEPTH ¦ % TRANSMISSION ¦ WATER TEMPERATURE	RECOVERY?	WATER DEPTH	DEPTH UNIT
						3 LET	4 LETTERS	5 LET/NO					Y/N	4 NO	LET
START/END ØF STATIØNS (ØPTIØNAL)	✓	162	1400	14	57	NAV	STAT	* 57		X		AR HYDRØ/GEØL/GEØP ST 57		* 40	M
	✓		1721			NAV	STAT	57			X	LV STATIØN 57		53	M
			1413									DREDGE ØVERSIDE – GØING DOWN			
START/END ØF SAMPLE COLLECTED OVER DISTANCE INTERVAL	✓		1417			SAM	PIPE	57P1		X		BIG TUG - START SAMPLING		40	M
	✓		1501			SAM	PIPE	57P1			X	WIRE CØMING UP	Y	33	M
SAMPLE COLLECTED AT DISCRETE LOCATION	✓		1533			SAM	DART	57D1	X			CØRE ØN BØTTM-BENT BARRL	Y	45	M
			1539									WIRE SNAG ØN A-FRAME			
			1543									DART ØN DECK SAMPLE DISTURBED			
			1610									BOTTLE IN WATER – GØING DØWN			
WATER SAMPLE	✓		1612			SAM	NANS	31W1	X			SFC H2Ø SAMP. SFC T=12°C	Y	42	M
RECORDING TRANSMISSOMETER NUMERICAL OBSERVATIØNS	✓		1630			ØBS	TRAN	31TA	X			1 METER ¦ 83 % ¦ 12°C	Y	42	M
	✓		1632					31TB	X			10 METR ¦ 42 % ¦ 7°C	Y	42	M
	✓		1635					31TC	X			20 METR ¦ 18 % ¦ 4.5°C	Y	42	M
DEPLOYMENT ØF: 1. SØNØBUØY	✓		1650			ØBS	SØNØ	1		X		SØNØBUØY # 1 ØVERBØARD		48	M
2. EXPLØSIVE CHARGE	✓		1712			ØBS	BØØM	3	X			EXPLØSIVE # 3 ØVERBØARD		51	M
		1-3	5-8	12-15	19-19	21-23	25-28	29-33	36	37	38	41-64	66	67-70	71

REMARKS: EQUIP., OPERATION, WEATHER, ETC.

≈ 1539: SEVERE DAMAGE TØ DART CORE – WINCH FAILURE CAUSED WIRE JAMMING IN A-FRAME

✗ 1721: EXPLOSIVE CAUGHT IN TRAILING SEAWEED; BLEW HOLE IN FANTAIL, SINKING FAST.

* PLEASE RIGHT-JUSTIFY THESE ENTRIES

** FORMAT FOR SAMPLE LABEL IS CRUISE LOCATOR-SAMPLE ATTEMPT # (E.G. S2-78-103)

FIGURE 9 EXAMPLE OF HOW TO FILL OUT STATION OPERATIONS LOG (see page 30)

d. A check in the "Punch Flag" column (left-most column of the logsheet both for entries outlined in 2c. above and for start/end times of stations if they are recorded in this log.

3. "Desirable" logbook entries - discretion of Chief Scientist:

a. Header information (ship, lat., long., etc.)

b. Sampling operation information such as:

1) Times of sample device deployment/recovery

2) Course/speed, wire out, wire angle, etc.

3) Start/end times for coming on and leaving a "station" where multiple sampling attempts are made (e.g., numbering system A -- see part II.D. of this section)

4) Data values such as current meter readings; depths of water samples in the water column, etc. (to be listed in the "Comments" field)

E. <u>Cruise Data Index Log</u> (CDI log) (see page 34 for example)

1. General rules

a. The CDI log will be keypunched in its entirety. Be neat and print legibly with block letters.

b. Logbook entries may be made in any order or at any time.

c. In general, the only entries made in this log by the watchstanders will be for tracklines and navigation data (see 2b. below). The navigator will be responsible for these entries.

2. "Required" logsheet entries (*Entries described below and marked by '*', need not be entered in the CDI log if entered in the Acoustic, Geopotential, or Station Operations logs)

a. Cruise Locator

b. Start/end time for:

1) Tracklines - trackline numbers in this case are entered in the "Sequence" field but may also be entered in the "Line No." field. "Spec." and "Data" codes used are: NAV and TRAK.

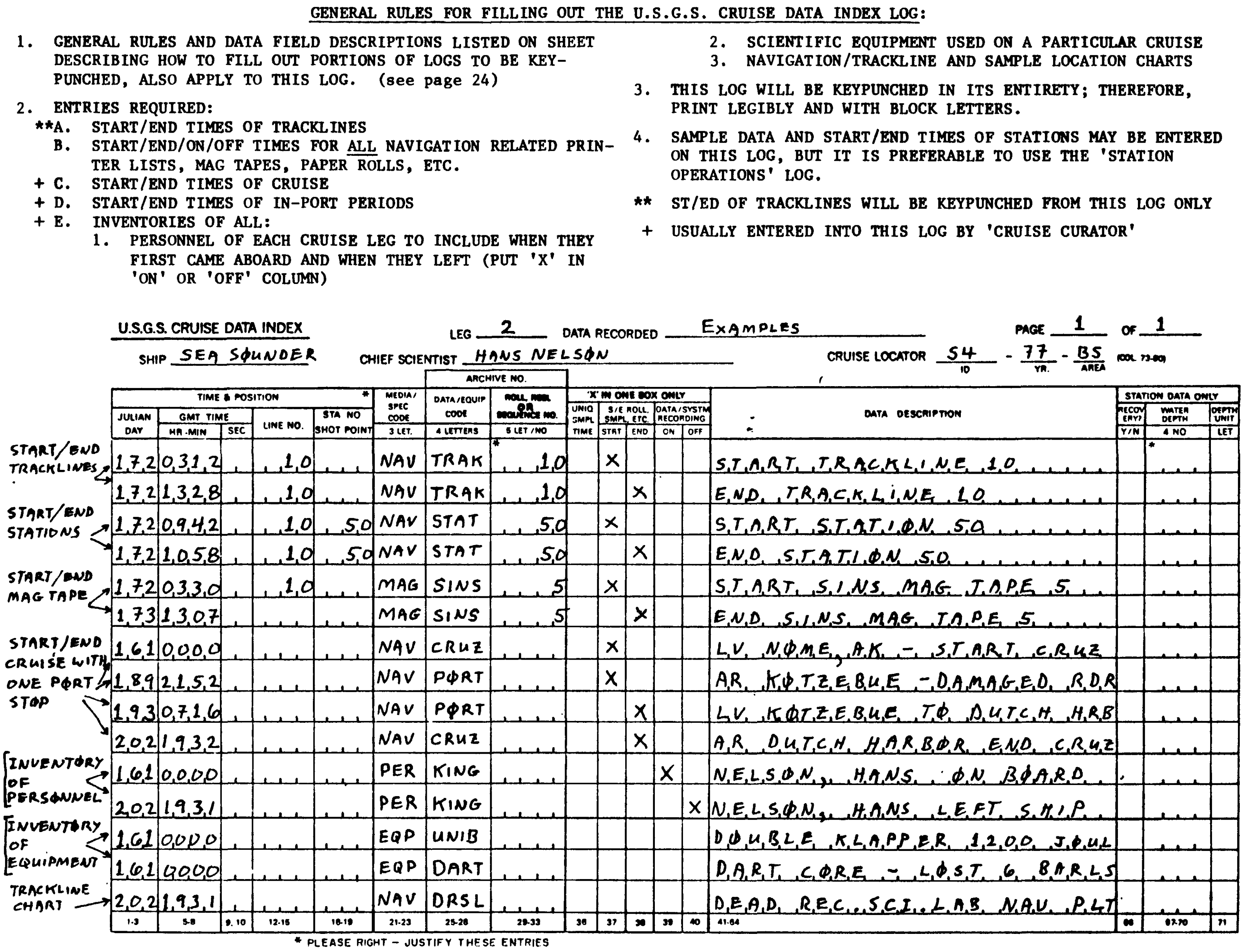
*2) Analog records and printer listings pertaining to navigation data.

*3) Magnetic and paper tapes pertaining to navigation data.

*4) Periods during which equipment that is part of multi-instrument system is in operation

3. "Desirable" logsheet entries - to be entered at the discretion of the Chief Scientist

   a. Start/end times for coming on and leaving a station. However, it is preferable to enter these times only on the "Station Operations" log if it is to be entered at all. "Data Code" used is STAT.



GENERAL RULES FOR FILLING OUT THE U.S.G.S. CRUISE DATA INDEX LOG:

1. GENERAL RULES AND DATA FIELD DESCRIPTIONS LISTED ON SHEET DESCRIBING HOW TO FILL OUT PORTIONS OF LOGS TO BE KEYPUNCHED, ALSO APPLY TO THIS LOG. (see page 24)

2. ENTRIES REQUIRED:
   **A. START/END TIMES OF TRACKLINES
   B. START/END/ON/OFF TIMES FOR ALL NAVIGATION RELATED PRINTER LISTS, MAG TAPES, PAPER ROLLS, ETC.
   + C. START/END TIMES OF CRUISE
   + D. START/END TIMES OF IN-PORT PERIODS
   + E. INVENTORIES OF ALL:
   1. PERSONNEL OF EACH CRUISE LEG TO INCLUDE WHEN THEY FIRST CAME ABOARD AND WHEN THEY LEFT (PUT 'X' IN 'ON' OR 'OFF' COLUMN)
   2. SCIENTIFIC EQUIPMENT USED ON A PARTICULAR CRUISE
   3. NAVIGATION/TRACKLINE AND SAMPLE LOCATION CHARTS

3. THIS LOG WILL BE KEYPUNCHED IN ITS ENTIRETY; THEREFORE, PRINT LEGIBLY AND WITH BLOCK LETTERS.

4. SAMPLE DATA AND START/END TIMES OF STATIONS MAY BE ENTERED ON THIS LOG, BUT IT IS PREFERABLE TO USE THE 'STATION OPERATIONS' LOG.

** ST/ED OF TRACKLINES WILL BE KEYPUNCHED FROM THIS LOG ONLY

+ USUALLY ENTERED INTO THIS LOG BY 'CRUISE CURATOR'

U.S.G.S. CRUISE DATA INDEX

LEG 2 DATA RECORDED EXAMPLES PAGE 1 OF 1

SHIP SEA SØUNDER CHIEF SCIENTIST HANS NELSØN CRUISE LOCATOR 54 - 77 - BS (COL 73-80)
ID YR. AREA

	TIME & POSITION *					MEDIA/ SPEC CODE	ARCHIVE NO.		'X' IN ONE BOX ONLY						STATION DATA ONLY		
	JULIAN DAY	GMT TIME HR-MIN	SEC	LINE NO.	STA NO SHOT POINT	3 LET.	DATA/EQUIP CODE 4 LETTERS	ROLL, REEL OR SEQUENCE NO. 5 LET/NO	UNIQ SMPL TIME	S/E ROLL, SMPL, ETC. STRT	S/E ROLL, SMPL, ETC. END	DATA/SYSTM RECORDING ON	DATA/SYSTM RECORDING OFF	DATA DESCRIPTION	RECOVERY Y/N	WATER DEPTH 4 NO	DEPTH UNIT LET
START/END TRACKLINES	172	0312		10		NAV	TRAK	* 10		X				START TRACKLINE 10		*	
	172	1328		10		NAV	TRAK	10			X			END TRACKLINE 10			
START/END STATIONS	172	0942		10	50	NAV	STAT	50		X				START STATIØN 50			
	172	1058		10	50	NAV	STAT	50			X			END STATIØN 50			
START/END MAG TAPE	172	0330		10		MAG	SINS	5		X				START SINS MAG TAPE 5			
	173	1307				MAG	SINS	5			X			END SINS MAG TAPE 5			
START/END CRUISE WITH ONE PØRT STØP	161	0000				NAV	CRUZ			X				LV NØME, AK - START CRUZ			
	189	2152				NAV	PØRT			X				AR KØTZEBUE - DAMAGED RDR			
	193	0716				NAV	PØRT				X			LV KØTZEBUE TØ DUTCH HRB			
	202	1932				NAV	CRUZ				X			AR DUTCH HARBØR END CRUZ			
INVENTØRY OF PERSØNNEL	161	0000				PER	KING					X		NELSØN, HANS ØN BØARD			
	202	1931				PER	KING						X	NELSØN, HANS LEFT SHIP			
INVENTØRY OF EQUIPMENT	161	0000				EQP	UNIB							DØUBLE KLAPPER 1200 JØUL			
	161	0000				EQP	DART							DART CØRE - LØST 6 BARLS			
TRACKLINE CHART	202	1931				NAV	DRSL							DEAD REC., SCI. LAB NAV. PLT			
	1-3	5-8	9.10	12-15	16-19	21-23	25-28	29-33	36	37	38	39	40	41-64	66	67-70	71

* PLEASE RIGHT - JUSTIFY THESE ENTRIES

FIGURE 10 EXAMPLE OF HOW TO FILL OUT CRUISE DATA INDEX LOG (see page 32)

IV. ANNOTATION OF ANALOG RECORDS (continuous trace recordings on paper rolls)

A. General rules for all records

1. "Start/End Roll" labels or stamps must be used at each end of the analog record, and should be placed on the same side of the paper as the recorded data.
2. A "Start/End of Line" label or stamp (also called a "Routine Annotation" label or stamp) should be placed directly before and directly after breaks in the recorded data in order to indicate the type of data break and the time of its occurrence. Breaks in data recording intervals take place at:
   a. Trackline changes
   b. Equipment failures affecting the data recording system.

   For each of these types of interuption of the recorded data, the paper on the analog recorder used for low resolution sparker or airgun data must be advanced a sufficient distance to allow the label or stamp to be used. Two types of "Start/End of Line - Routine Annotation" labels and stamps are available as shown on page 36. Type A is generally used only with low resolution seismic systems and at the discretion of the Chief Scientist.
3. All analog records must be annotated carefully and with large letters (using either a "Routine Annotation" label/stamp or by hand printing): (see pages 36 and 37)
   a. At least once each hour
      1) MDIG "required" information: day, time, and recorder scale.
      2) Any other information required by Chief Scientist: e.g., line, course, speed, fire rate, sweep rate, power, filter settings, water depth, etc.

   b. Whenever there is a change in the recording system that affects the appearance of the data (e.g., scale change, filter settings, fire/sweep rate). Both the time and the change made must be noted on the record.
   c. Depending upon recorder sweep rate, it may be desirable to mark time on the records more frequently than once each hour -- at the discretion of the Chief Scientist.

USE OF LABELS/STAMPS FOR ANNOTATING

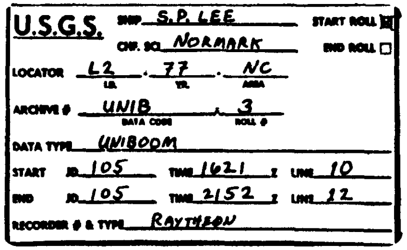
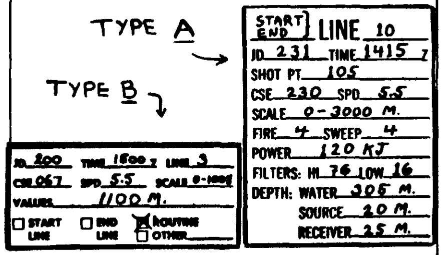
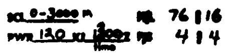
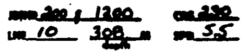
NAME OF STAMP	START/END OF ROLL	START/END OF LINE and ROUTINE ANNOTATION	RAYTHEON ANNOTATION ('RAYTHEON' SEISMIC)	
EXAMPLE	 U.S.G.S. SHIP S.P. LEE START ROLL ☒ CHF. SCI. NORMARK END ROLL □ LOCATOR L2 . 77 . NC (LR. YR. AREA) ARCHIVE # UNIB . 3 (DATA CODE ROLL #) DATA TYPE UNIBOOM START JD 105 TIME 1621 Z LINE 10 END JD 105 TIME 2152 Z LINE 12 RECORDER # & TYPE RAYTHEON	 TYPE A START/END LINE 10 JD 231 TIME 1415 Z SHOT PT 105 CSE 230 SPD 5.5 SCALE 0-3000 M. FIRE 4 SWEEP 4 POWER 120 KJ FILTERS: HI 76 LOW 16 DEPTH: WATER 305 M. SOURCE 20 M. RECEIVER 25 M. TYPE B JD 200 TIME 1500 Z LINE 3 CSE 067 SPD 5.5 SCALE 0-1000 VALUES 1100 M. □ START LINE □ END LINE ☒ ROUTINE □ OTHER	I  SCL 0-3000 m HI 76 16 PWR 130 KJ 1200 Z LO 4 4 *****Use of these stamps/labels is optional - see pages 21, 23, and 27 for 'required' and 'desirable' information.	II  10 308 m CSE 230 SPD 5.5
WHERE TO USE	Use at start and end of all Analog Records	Use on all Analog Records (A or B)	'Raytheon' Seismic record	'Raytheon' Seismic record
WHEN TO USE	Use 2 times per Analog Record[1]	1)Start/End of lines[2] (A or B) 2)Hourly record annotation and whenever there is a change in the recorder system (except 'Raytheon' seismic records)[3] (TYPE B ONLY) 3)Equipment failures[2] (A or B)	Routinely on the odd hours[4] Use appropriate stamp whenever there is a change in the recorder system	Routinely on the even hours[4]
INSTRUCTIONS	[1]Mark the inner side of the record (e.g. SAME SIDE AS DATA)	[2] Pull out enough paper to create a gap in record for the use of this stamp. [3] IMPORTANT Draw an arrow from the stamp to the corresponding time mark.	[4] IMPORTANT Draw an arrow from the stamp to the corresponding time mark.	

FIGURE 11 EXAMPLE OF ANNOTATION LABELS AND STAMPS (see page 35)

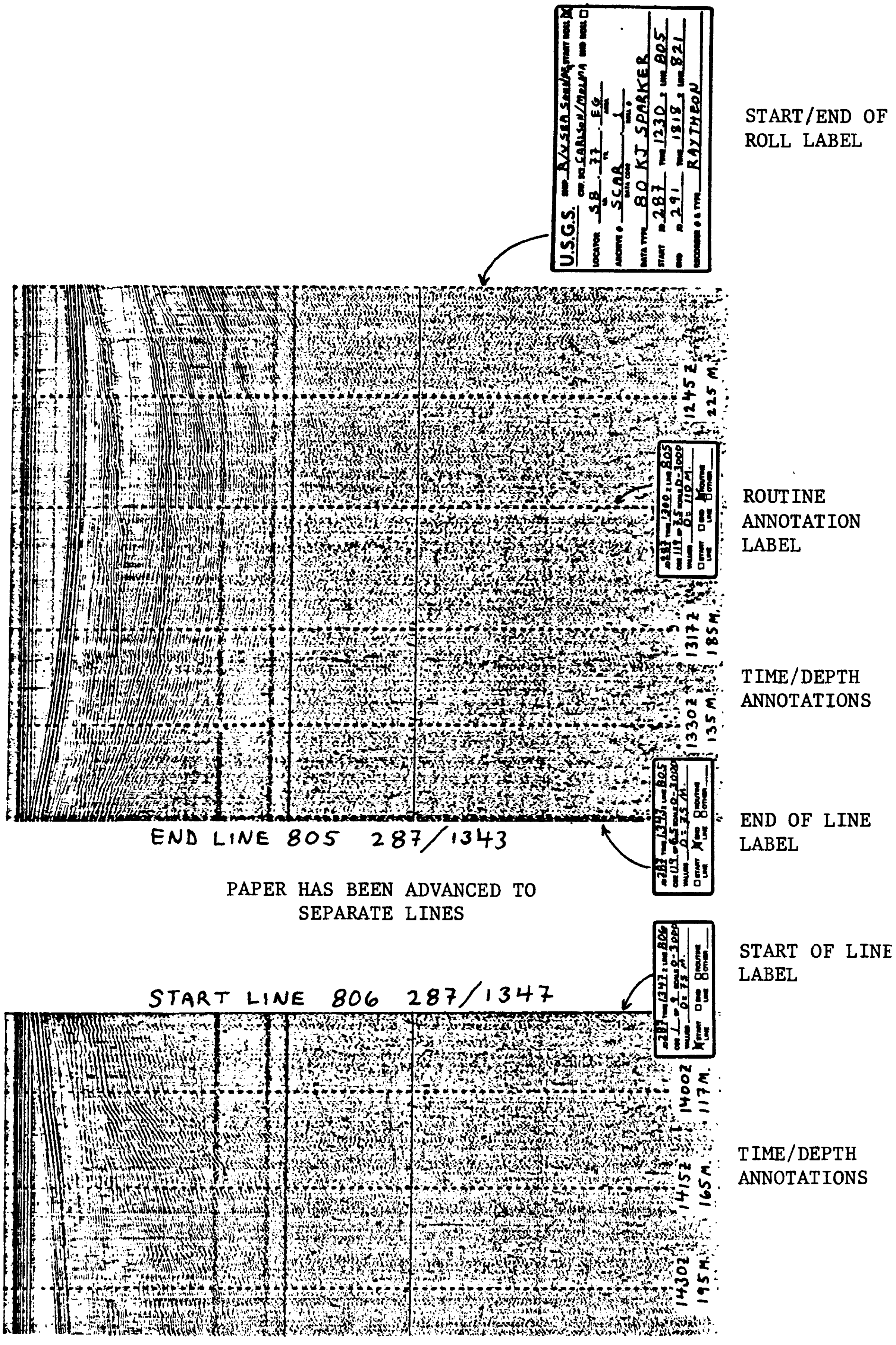


FIGURE 12 EXAMPLE OF RECORD ANNOTATION AND LABELING (SEE PAGE 35)

V. LABELLING OF SAMPLES

A. All samples must be assigned a unique "Sample No.", and this number must be affixed to the sample or sample container. The format for the "Sample No." must be:

Cruise Locator - "Sampling Attempt" No.

(e.g., S4 - 76 - 101) See part II. C. and D., pages 19 and 20 for definitions and numbering systems.

B. Use one of the "physical sample" gummed labels shown on page 40 or their equivalent (see example below)

1. Required information:

   a. "Sample Number" (Cruise Locator + Sampling Attempt No.)

   b. Four-letter "Data/Equipment Code" that designates the type of sampling device used

2. Desirable information:

   a. Two-letter "area code"

   b. "Station No.", if different from the "Sample Attempt No." as in numbering system B (see page 20)

Example of Sample Label:

S4-76-101	:	DART
Sample No.		Equip. Code

VI. LABELS FOR ALL PHYSICAL DATA

A. General rules for use of labels

1. All physical cruise data must be labeled (written or gummed labels) with the "Cruise Locator" and optional two letter "area code".
2. Any physical cruise data which will eventually be stored in permanent archives must also be labeled (written or gummed labels) with the "Archive Number" or the "Sample Number" (see pages 18 & 20).
3. Labels should be placed on the physical data such that they are visible when the data is stored in its shipboard container (i.e., end of analog boxes, outside and inside of maps, front of sample bags, etc.)

B. Analog record boxes/bags

1. "Analog box" gummed label

C. Magnetic tape reels

1. "Mag tape reel" gummed label

D. Magnetic tape boxes

1. "Mag tape box" gummed label

E. Physical Station Samples

1. "Physical sample" gummed label

F. All Other Physical Data

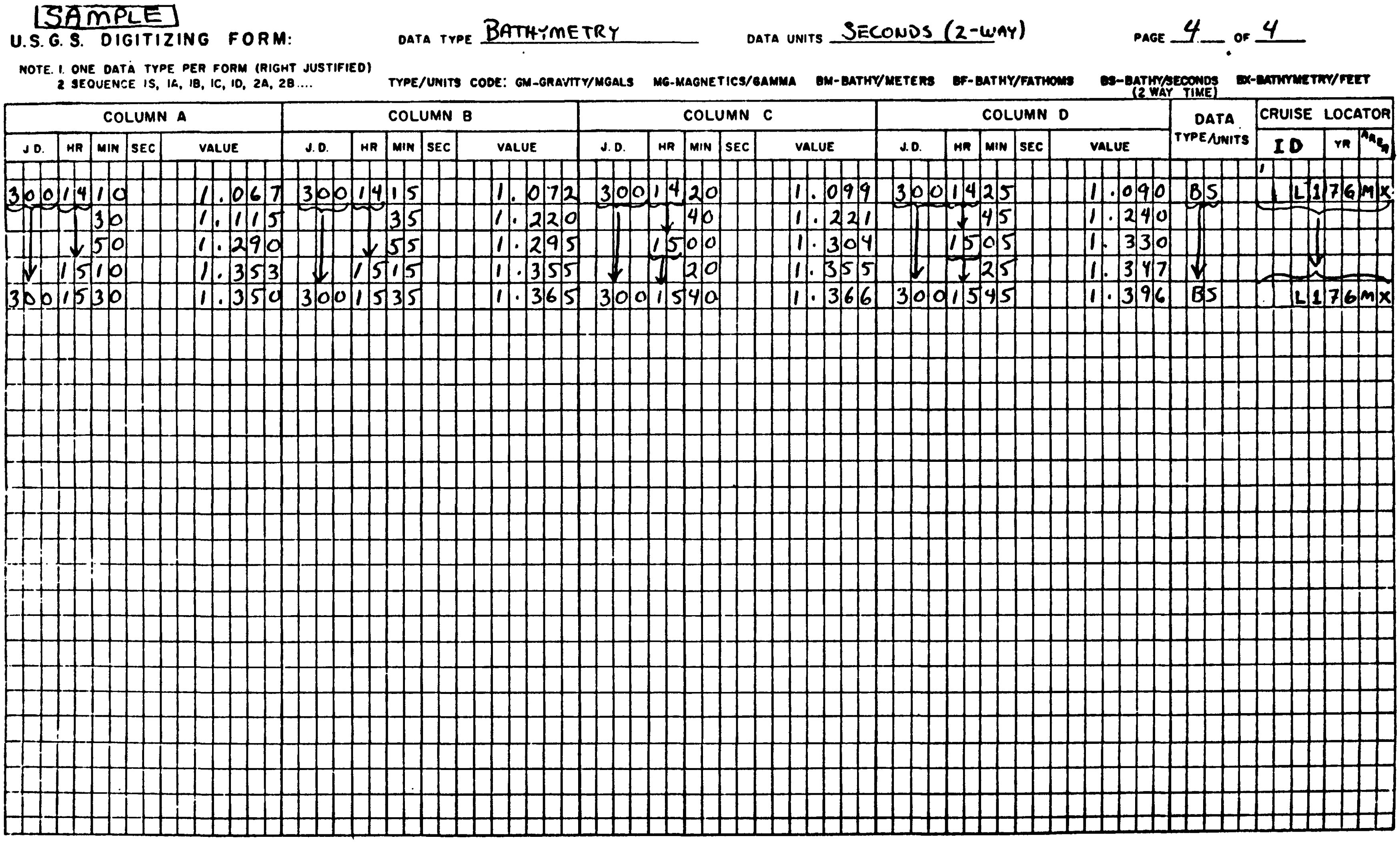
1. "Archive" gummed label

USE OF GUMMED LABELS

NAME OF LABEL	MAG. TAPE REEL	MAG. TAPE BOX	ANALOG BOX	PHYSICAL SAMPLE
EXAMPLE	U.S. GEOLOGICAL SURVEY LOCATOR L3 . 77 . NC I.D. YR AREA ARCHIVE # GRAD . 3 DATA CODE REEL# START: 300 . 0105 . 3 END: 301 . 1235 . 5 J.D. TIME(Z) LINE	USGS LOCATOR L 3 I 7 7 I MX ARCHIVE # GRAD I 3	LOC. L3 77 NC ARCH.# GRYS 7 START 310 1930 9 END 311 0340 10 J.D. TIME Z LINE	USGS/MARINE GEOLOGY LEG CORE SECTION SAMPLE INTERVAL L3-77 D7 THIS ↑ OR ↓ THIS USGS ARCH.# DART . 7 DATA CODE STN.# LINE # 10 SHOT# INIT. A.C. LOC. L3 . 77 . NC I.D. YR AREA
INSTRUCTIONS	Place on the front (clear plastic) side of all magnetic tape reels	Place on the side of all magnetic tape: 1) Cannisters or retainer straps 2) Storage boxes	Place on the end of all analog record storage boxes or place on the outside of plastic record bags	1) Place on all storage containers for physical station samples 2) Place on all misc. physical cruise data (e.g. maps, plots, etc.) Change "sta. # " to seq. # "

FIGURE 13 EXAMPLE OF HOW TO USE GUMMED LABELS

THIS LOG FORM IS AVAILABLE FOR CODING ANY KIND OF DIGITAL DATA THAT MAY BE COLLECTED ON A CRUISE. TYPICAL DATA MIGHT INCLUDE BATHYMETRY VALUES, CURRENT METER READINGS, OR GEOPOTENTIAL VALUES, ETC.



SAMPLE

U.S.G.S. DIGITIZING FORM: DATA TYPE Bathymetry DATA UNITS Seconds (2-way) PAGE 4 OF 4

NOTE. 1. ONE DATA TYPE PER FORM (RIGHT JUSTIFIED)
2 SEQUENCE 1S, 1A, 1B, 1C, 1D, 2A, 2B....

TYPE/UNITS CODE: GM-GRAVITY/MGALS MG-MAGNETICS/GAMMA BM-BATHY/METERS BF-BATHY/FATHOMS BS-BATHY/SECONDS (2 WAY TIME) BX-BATHYMETRY/FEET

COLUMN A					COLUMN B					COLUMN C					COLUMN D					DATA TYPE/UNITS	CRUISE LOCATOR		
J.D.	HR	MIN	SEC	VALUE	J.D.	HR	MIN	SEC	VALUE	J.D.	HR	MIN	SEC	VALUE	J.D.	HR	MIN	SEC	VALUE		ID	YR	AREA
300	14	10		1.067	300	14	15		1.072	300	14	20		1.099	300	14	25		1.090	BS	L1	76	MX
		30		1.115			35		1.220			40		1.221			45		1.240				
		50		1.290			55		1.295		15	00		1.304		15	05		1.330				
	15	10		1.353		15	15		1.355			20		1.355			25		1.347				
300	15	30		1.350	300	15	35		1.365	300	15	40		1.366	300	15	45		1.396	BS	L1	76	MX

FIGURE 14 EXAMPLE OF HOW TO USE DIGITIZING FORM

SECTION THREE

SPECIAL CONSIDERATIONS FOR NAVIGATION/GRAVITY/MAGNETICS/BATHYMETRY WATCHES

(SUMMARY)

I. WATCH CHANGE CHECKLIST

A. WWV time, Systron-Donner time, Sat. Nav. time displayed on the cathode ray tube (CRT) and on the printer listing.

B. Doppler sonar mode switch

C. Thumb wheels on gravimeter

D. Paper tape print-out on gravimeter

E. Line number entries on gravimeter and gradiometer

F. Gradiometer tuning

G. Velocimeter value on Sat. Nav. printout.

H. Water depths for gradiometer use

I. Way-points on chart and in Sat. Nav. system

J. Cesium clock settings

K. Mini-ranger log

L. Velocimeter water discharge

M. Turns on g-meter 3rd axis

N. Geopotential log

II. NECESSARY LOG ENTRIES OR PLOTTING

A. Every half hour

B. Events as they occur

SPECIAL CONSIDERATIONS FOR NAVIGATION/GRAVITY/MAGNETICS/BATHYMETRY WATCHES

(DETAIL)

I. WATCH CHANGE CHECKLIST - pertains primarily to navigation/geopotential systems on the S. P. LEE.

A. Compare time from WWV receiver with time on Systron-Donner clock and reset or adjust as necessary. Compare Sat. Nav. system CRT and printer time with corrected Systron-Donner time. Correct Sat. Nav. time by issuing a ST command and inputting a correct even minute (0, 2, 4,---58), hitting the RETURN key on 00 seconds. Compare gravity meter clock with Systron-Donner and reset if more than 2 seconds off.

B. Check the water depth. If less than 300 meters, mode switch on doppler sonar should be in the bottom track position. If greater than 300 meters and if the system is searching for bottom, mode switch should be in water track. Examine the trackline for your watch and get a feeling for the times when the doppler should be in each mode.

C. Check the thumbwheels on the gravimeter and make sure they are all in the proper positions.

D. Examine the paper tape printout on the gravimeter and verify that the data is correct. This assures that the data is going to tape correctly.

E. Check the line number entry on the gradiometer.

F. Check the tuning on the gradiometer by setting the Meter switch to Signal and adjusting the tuning switch for maximum deflection.

G. Check the "vson" value on the Sat. Nav. printout. It should be about 1500 m/sec. If it is 1400.0, the velocimeter is malfunctioning and should be checked. Calculate an approximate "vson" from the water temperature (and a salinity measurement if you are not in blue water).

H. Examine the trackline for your watch and determine if there will be any areas with water depths of less than 60-meters. The gradiometer will have to be shortened to 500 ft. in those areas.

I. Check the way-points on the chart against those entered in system. Check with the bridge and the seismic lab and verify that they have or know the significance of your way-points.

J. Check the meter readings on the cesium clock. Notify an ET immediately if any readings are out of spec.

K. Check the Mini-ranger log if that system is being used and verify that the correct station number, locations and channel numbers are entered.

L. Check that the velocimeter is discharging water from the port side.

M. Check on the number of wraps on the gravimeter 3rd axis.

N. Completely enter all pertinent geopotential log values and draw line for the repeated values. (see pages 27-29)

II. NECESSARY LOG ENTRIES OR PLOTTING

A. Every half hour (see pages 23,25,27-28, and 29 for more detail)

1. Bathymetry log entry: Day/time; course/speed/ water depth; line number.
2. Gravity log entry: Day/time; course/speed; line number; analog gravity value; spring tension; check digital gravity.
3. Gradiometer log entry: Day/time; course/speed; line number; master/slave values.
4. Navigation: plot fix.

B. Events as they occur

1. Log all satellite fixes (except those that are purged) and their standard deviations.
2. Make log entries for whenever individual navigation systems (i.e., doppler sonar, Loran-C, Sat.-Nav., etc.) start up or shut down.
3. Change in DR standard deviation to force solution onto satellite or Loran-C
4. Change in MiniRanger station location and/or channel identification.

SECTION FOUR

SPECIAL LOGFORMS FOR SEDIMENTS, SONOBUOYS AND GRAVITY LAND TIES

I. FORMS FOR SEDIMENT SAMPLES - The series of forms described below are designed to record and archive general data about any sample taken. These forms will be xeroxed and the chief scientists will have a copy. The archive copy will be put together with an X-radiograph (if appropriate) and a photograph of the sample so that a library can be maintained which will allow anyone to sit at one location and read about the description of the sample, what other things were attempted at that station, and where that sample has been subsampled, for what purpose, and by whom. The forms will be provided to the ship and are designed to help the chief scientists in their initial examinations of the samples. Thus, these forms should not require any extra burden on you and hopefully, you will learn to love them.

A. Sample Log (see example, page 47)

This form is provided so that a permanent record of subsamples taken from each sediment sample is kept. Terms used on the form are describe below.

1. Section: The 150 cm section from which the subsample was taken. For box cores, van veens, etc., this column is blank.
2. Interval: This is the interval within the 150 cm section where the sample was taken (i.e., Section 2, 225-227 cm)
3. Name: The name of the person for which the sample was taken.
4. Organization: The address of the above person so that others can contact them.
5. Purpose: Why the sample was taken (i.e., grain-size, clay mineralogy, forams, etc.) This lets others know what work has been done on a core.
6. Date: When the sample was taken.

B. Visual Core Description (see examples, page 48)

This form is provided so that megascopic descriptions of cores are recorded in a uniform fashion. No other form should be used for this purpose.

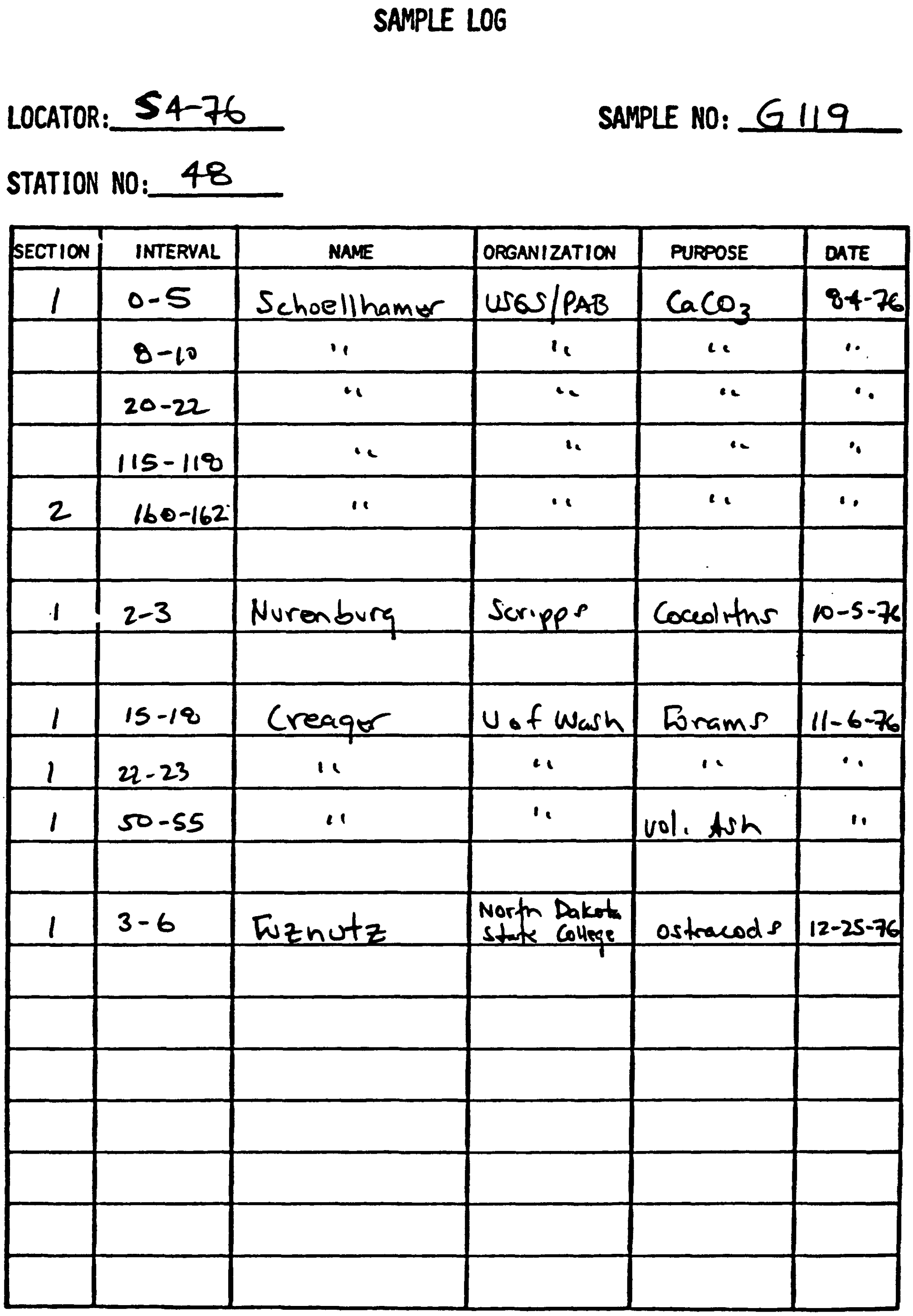
1. Graphic representation: This column should be used to draw in features such as sedimentary structures, fossils, contacts, etc., at the appropriate depths and the bottom of the core. For cores longer than 150 cm, additional sheets are used and the appropriate section number is recorded in the top-right box.
2. Smear slide and sub-samples: Place an asterisk (*) and the depth where each recorded smear slide is taken. You can also place a symbol (T - Texture, M - Mineralogy, etc.) where subsamples have been removed for those purposes.
3. Deformation: Indicate whether the sediment is soupy, firm, stiff, etc.
4. Large space to the right of the column: This space is provided for a detailed description of the core. The preferred order of description is shown below:

   Sediment name, underlined (determined from smear-slide examinations); Color of sediment, using both name and GSA Rock Color cahrt number; Induration, (soft, stiff, etc.); Structures (ripples, crossbeds, etc.); Burrowing, deformation, and then anything else worth recording. (see example)

C. Smear Slide Description (see example, page 49)

This form is provided so that smear-slide descriptions can be recorded easily. The preferred coding for abundance is shown below:

CODE	MEANING	RANGE
T	TRACE	< 1%
R	RARE	1-5%
C	COMMON	5-25%
A	ABUNDANT	25-75%
D	DOMINANT	> 75%



SAMPLE LOG

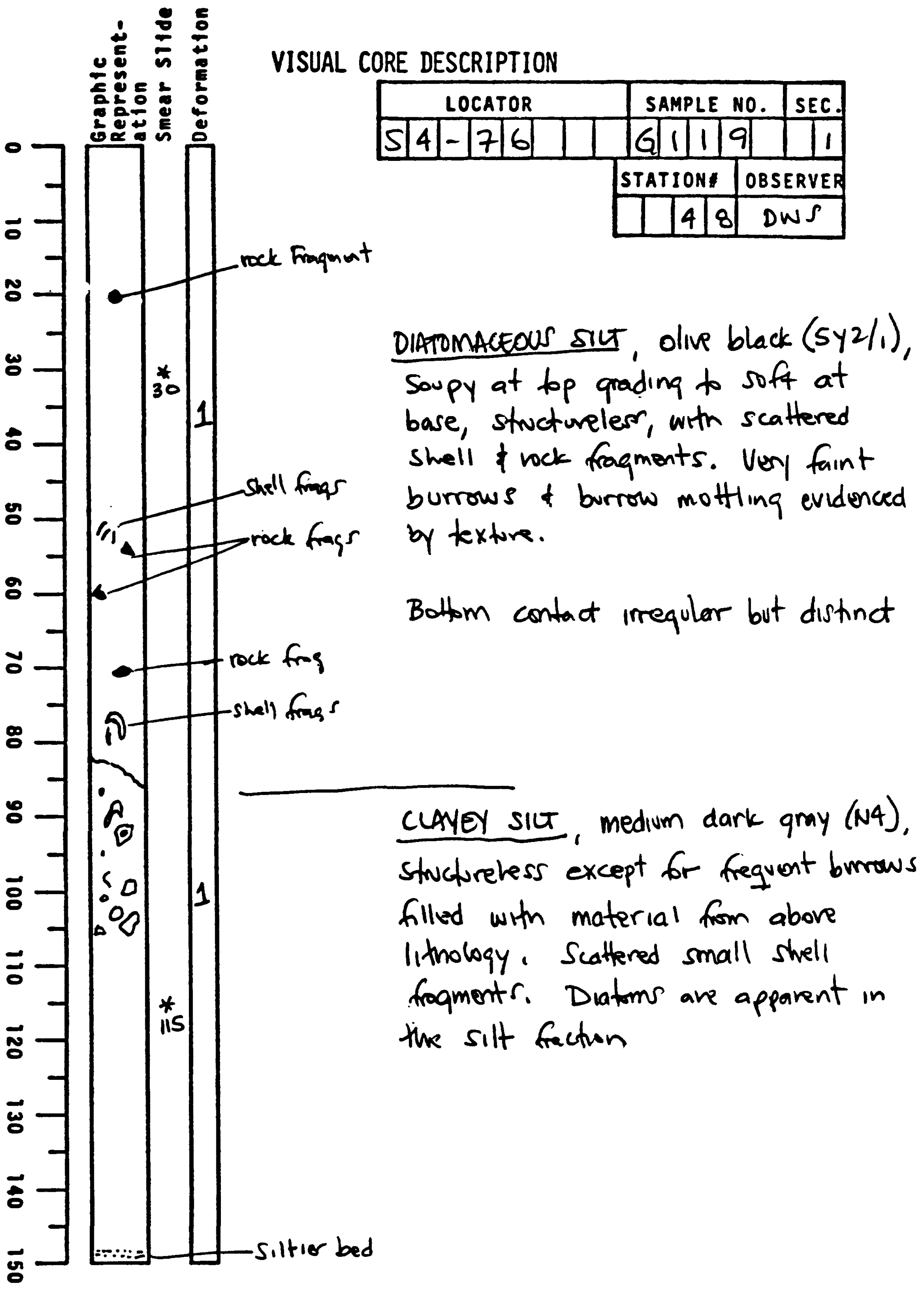
LOCATOR: S4-76

SAMPLE NO: G 119

STATION NO: 48

SECTION	INTERVAL	NAME	ORGANIZATION	PURPOSE	DATE
1	0-5	Schoellhamer	USGS/PAB	$CaCO_3$	8-4-76
	8-10	"	"	"	"
	20-22	"	"	"	"
	115-118	"	"	"	"
2	160-162	"	"	"	"
1	2-3	Nurenburg	Scripps	Coccoliths	10-5-76
1	15-18	Creager	U of Wash	Forams	11-6-76
1	22-23	"	"	"	"
1	50-55	"	"	vol. Ash	"
1	3-6	Fuznutz	North Dakota State College	ostracods	12-25-76

FIGURE 15 EXAMPLE OF HOW TO FILL OUT SUB-SAMPLE LOG (SEE PAGE 45)



Graphic Represent-ation
Smear Slide
Deformation
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
rock Fragment
*30
1
Shell frags
rock frags
rock frag
shell frags
1
*115
Siltier bed

VISUAL CORE DESCRIPTION

LOCATOR								SAMPLE NO.						SEC.
S	4	-	7	6				6	1	1	9			1

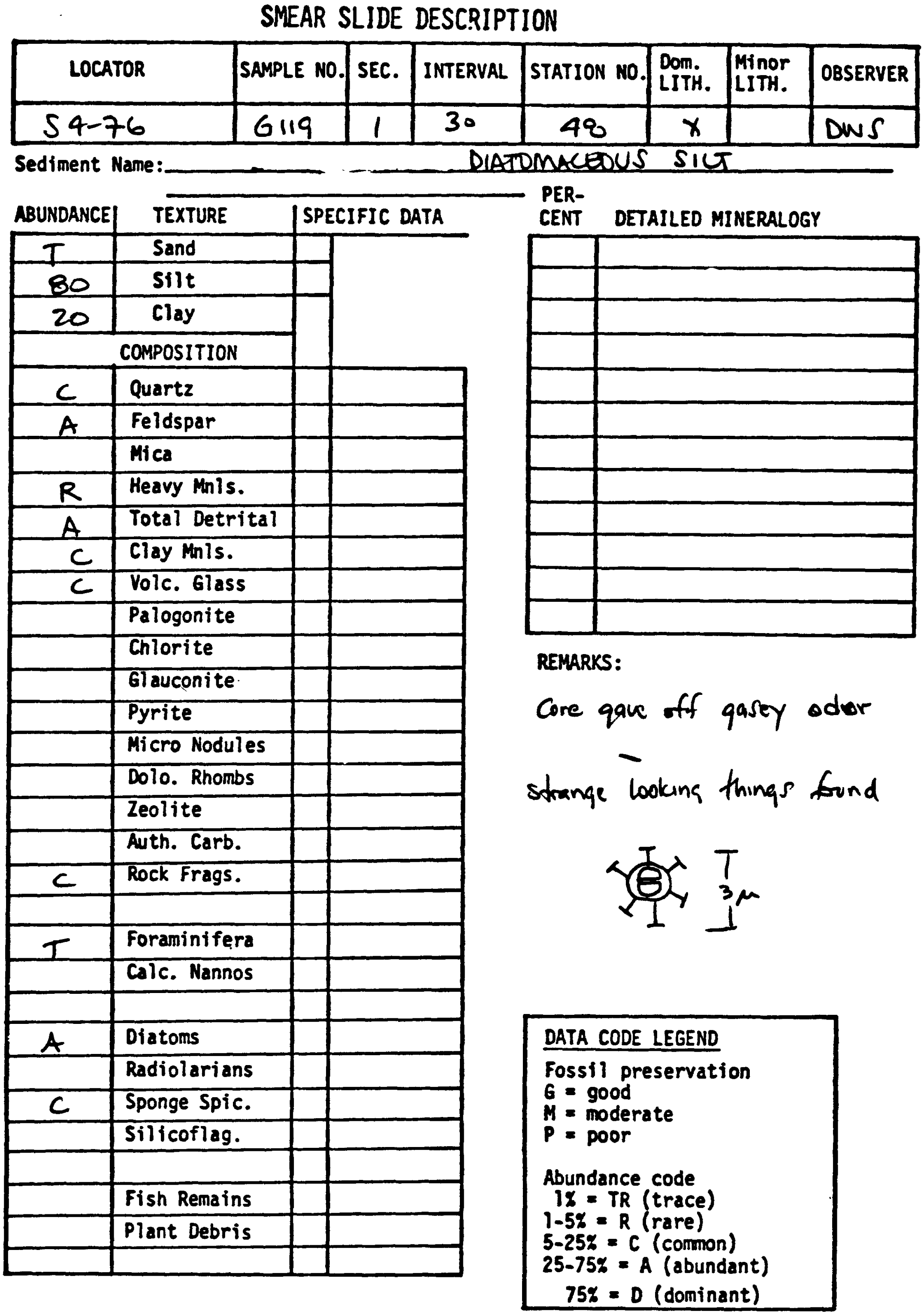
STATION#				OBSERVER
		4	8	DWS

DIATOMACEOUS SILT, olive black (5Y2/1), Soupy at top grading to soft at base, structureless, with scattered shell & rock fragments. Very faint burrows & burrow mottling evidenced by texture.

Bottom contact irregular but distinct

CLAYEY SILT, medium dark gray (N4), structureless except for frequent burrows filled with material from above lithology. Scattered small shell fragments. Diatoms are apparent in the silt fraction

FIGURE 16 EXAMPLE OF HOW TO FILL OUT VISUAL CORE DESCRIPTION LOG (SEE PAGES 45 and 46)



SMEAR SLIDE DESCRIPTION

LOCATOR	SAMPLE NO.	SEC.	INTERVAL	STATION NO.	Dom. LITH.	Minor LITH.	OBSERVER
S4-76	6119	1	30	48	X		DWS

Sediment Name: DIATOMACEOUS SILT

ABUNDANCE	TEXTURE	SPECIFIC DATA	
T	Sand		
80	Silt		
20	Clay		
	COMPOSITION		
C	Quartz		
A	Feldspar		
	Mica		
R	Heavy Mnls.		
A	Total Detrital		
C	Clay Mnls.		
C	Volc. Glass		
	Palogonite		
	Chlorite		
	Glauconite		
	Pyrite		
	Micro Nodules		
	Dolo. Rhombs		
	Zeolite		
	Auth. Carb.		
C	Rock Frags.		
T	Foraminifera		
	Calc. Nannos		
A	Diatoms		
	Radiolarians		
C	Sponge Spic.		
	Silicoflag.		
	Fish Remains		
	Plant Debris		

PER-CENT	DETAILED MINERALOGY

REMARKS:

Core gave off gasey odor

—

strange looking things found

3μ

DATA CODE LEGEND

Fossil preservation
G = good
M = moderate
P = poor

Abundance code
1% = TR (trace)
1-5% = R (rare)
5-25% = C (common)
25-75% = A (abundant)
75% = D (dominant)

FIGURE 17 EXAMPLE OF HOW TO FILL OUT SMEAR SLIDE DESCRIPTION LOG (SEE PAGE 46)

II. PROCEDURES AND FORMS FOR SONOBUOY OPERATIONS - A checklist of the procedures required for the preparation and launching of sonobuoys is included here so that the watchstander may become familiar with the items required on the two logsheets. The first logsheet, "Sonobuoy Station Log" (page 53), lists the technical information that is required both for processing the sonobuoy data and for making a claim for sonobuoy replacement, if the buoy fails. The second logsheet, "U.S.G.S. Station Operations Log" (page 54), gives a sequential listing of events that occured during the sonobuoy station. Some items in the "U.S.G.S. Station Operations Log" are "required" entries, because they are needed to produce the post-cruise computer data listings; these items are described in "C." below in both written and pictorial form.

A. Sonobuoy Preparation and Launch Checklist

1. Pre-launch check of sonobuoy
    a. If possible, start transmitting on deck to ensure that sonobuoy batteries are working and that the selected crystal frequency is correct.
    b. Check the hydrophone depth and 20 minute delay
2. Notify radio officer and bridge of intent to deploy buoy
3. Fill out log forms
4. Prepare analog tape recording system
    a. Voice introduction, including date, time, identification, etc.
    b. Set footage counter
    c. Check input levels on each channel
5. Start recording vertical incidence seismic information on tape and graphic recorders
6. Pre-launch lab check
    a. Ensure that the correct crystal is being used in the receiver and that all recorder inputs are coming from the receiver module being used.
    b. Ensure that the correct initial settings (gain, hi-cut filter, waterbreak gain) have been set on the receiver.
    c. Ensure that antenna pre-amp gain has been turned down (for low frequency sonobuoys).

        d. Check signal paths (seismic signal and triggers) to both graphic and analog tape recorders.

    7. Proceed with countdown to launch

        a. Radio operator on deck counts down airgun shots, and reports when the sonobuoy is in the water and adjacent to the airguns.

        b. Lab procedure:

            1) Record start of countdown on tape
            2) Set correct feed rates on recorders
            3) Set proper display mode on recorders
            4) Switch recorder input from vertical incidence seismic to sonobuoy seismic
            5) Mark the first trace (when sonobuoy at airguns) on the recorders
            6) Record time and footage reading in log and on tape recorder
            7) Re-tune the sonobuoy on the receiver
            8) Adjust the graphic recorders for optimum display

B. Sonobuoy Station Log

    1. All entries on this form are important

    2. Please fill in the information requested (see example, page 53)

C. U.S.G.S. Station Operations Log

    1. The "required" entries that must be written down either as they happen or after the last regular entry are:

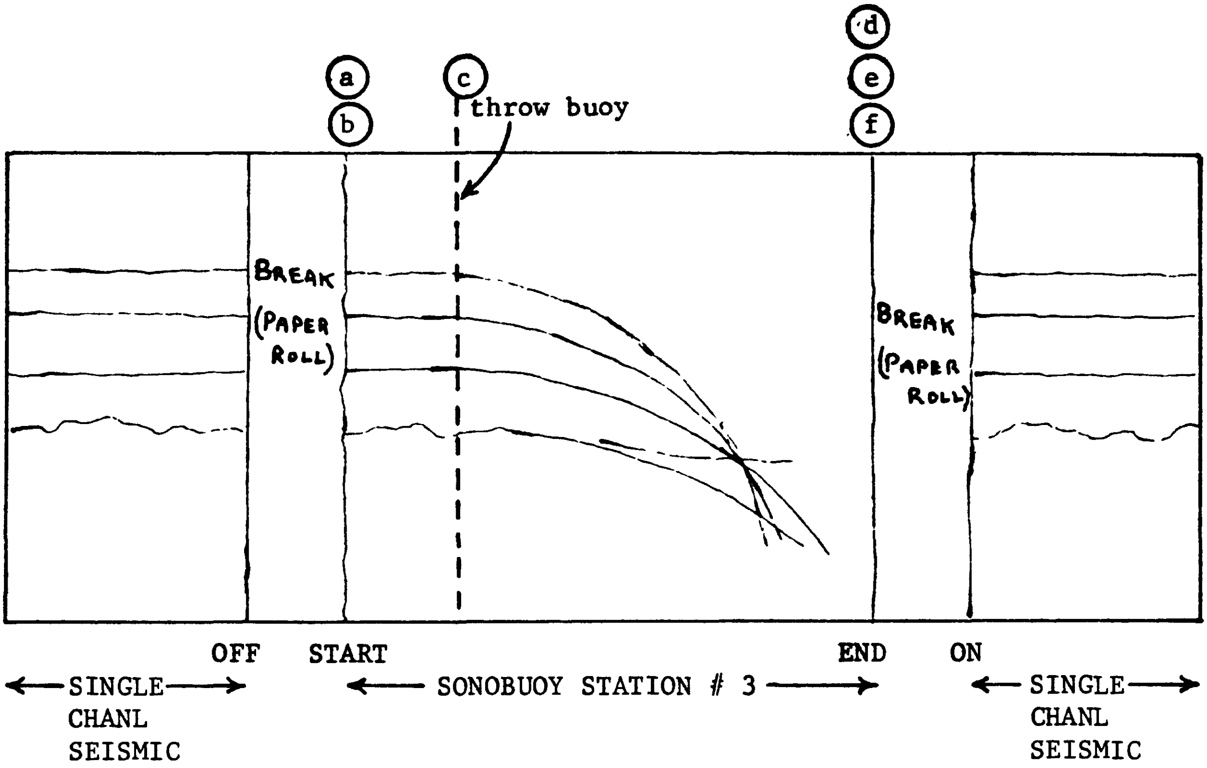
        a. PAP SONO 3 START
        b. MAG SONO 3 START } Denotes the time at which paper/mag tape start recording.

        c. OBS SONO 3 START – Denotes time at which sonobuoy # 3 is thrown into water.

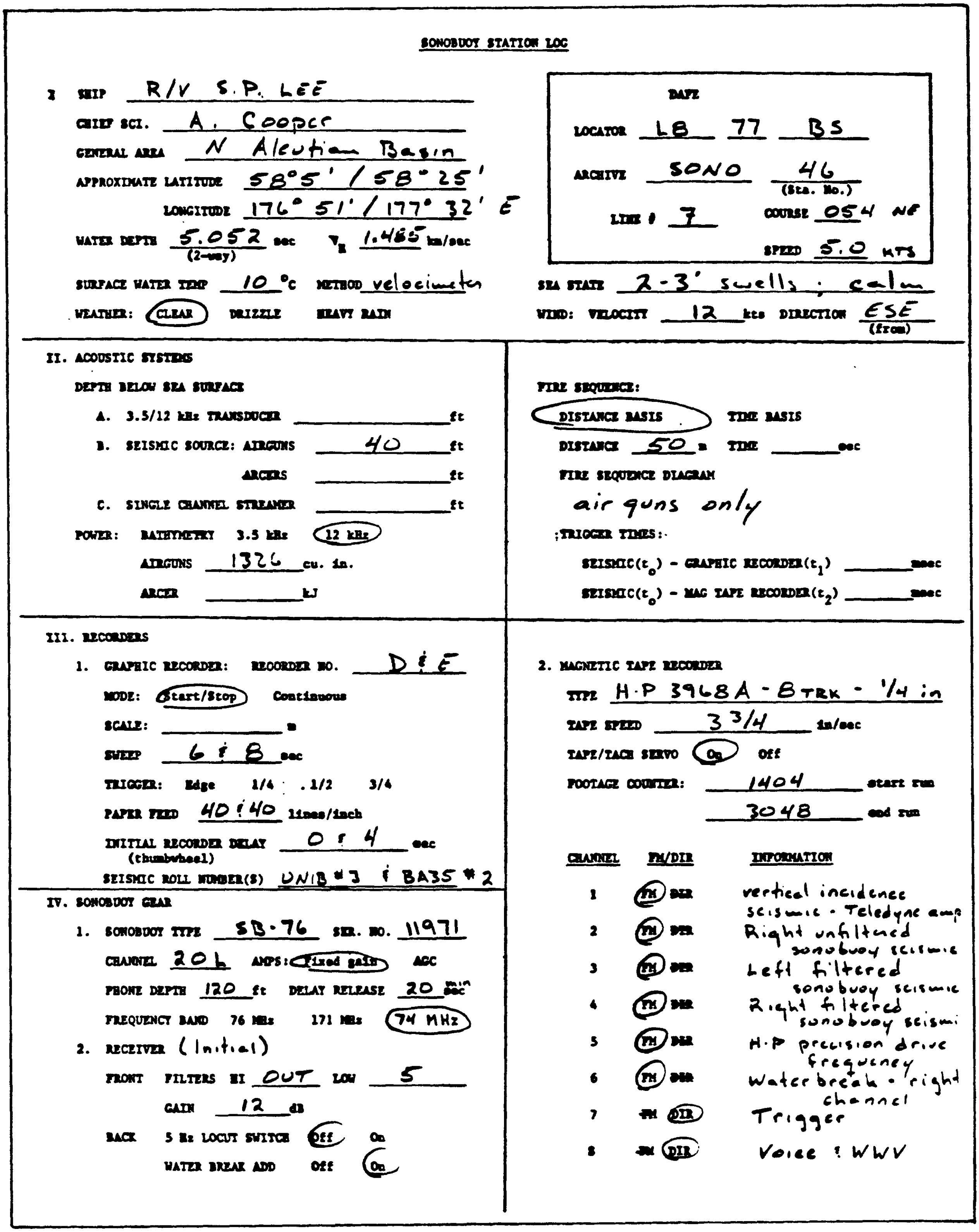
        d. OBS SONO 3 END – Denotes time at which sonobuoy #3 signal is lost.

        e. PAP SONO 3 END
        f. MAG SONO 3 END } Denotes time at which paper/mag tape for sonobuoy # 3 ends.

        **See diagram at top of following page; also note examples of logsheets on pages 54 and 55.



2. If more than one graphic recorder is being used, items "a." and "e." above must be repeated for each recorder.
   a. If two records of the same buoy are being made, distinguish the two records by adding a letter before the number (e.g., "3" and "S 3"). You can use any letter(s) in place of "S".
   b. If records of two different buoys are being made, distinguish the two records by two different numbers (e.g., "3" and "4").
3. If more than one buoy is in the water at the same time, items "1.c." and "1.d." above must be repeated for each buoy. Distinguish the two buoys by two different numbers (e.g., "3" and "4").
4. If more than one magnetic tape is used to record a single buoy, then items "1.b." and "1.f." above must be repeated for each reel of tape. Distinguish the two reels by adding a letter after the number (e.g., "3" and "3A").



SONOBUOY STATION LOG

I SHIP R/V S.P. LEE

CHIEF SCI. A. Cooper

GENERAL AREA N Aleutian Basin

APPROXIMATE LATITUDE 58°5' / 58°25'

LONGITUDE 176° 51' / 177° 32' E

WATER DEPTH 5.052 sec (2-way) V_R 1.485 km/sec

SURFACE WATER TEMP 10 °C METHOD velocimeter

WEATHER: [✓ CLEAR] DRIZZLE HEAVY RAIN

DATE

LOCATOR LB 77 BS

ARCHIVE SONO 46 (Sta. No.)

LINE # 7 COURSE 054 NE

SPEED 5.0 KTS

SEA STATE 2-3' swells ; calm

WIND: VELOCITY 12 kts DIRECTION ESE (from)

II. ACOUSTIC SYSTEMS

DEPTH BELOW SEA SURFACE

A. 3.5/12 kHz TRANSDUCER ______ ft

B. SEISMIC SOURCE: AIRGUNS 40 ft

ARCERS ______ ft

C. SINGLE CHANNEL STREAMER ______ ft

POWER: BATHYMETRY 3.5 kHz [✓ 12 kHz]

AIRGUNS 1326 cu. in.

ARCER ______ kJ

FIRE SEQUENCE:

[✓ DISTANCE BASIS] TIME BASIS

DISTANCE 50 m TIME ______ sec

FIRE SEQUENCE DIAGRAM

air guns only

;TRIGGER TIMES:

SEISMIC(t_0) - GRAPHIC RECORDER(t_1) ______ msec

SEISMIC(t_0) - MAG TAPE RECORDER(t_2) ______ msec

III. RECORDERS

1. GRAPHIC RECORDER: RECORDER NO. D & E

MODE: [✓ Start/Stop] Continuous

SCALE: ______ m

SWEEP 6 & 8 sec

TRIGGER: Edge 1/4 . 1/2 3/4

PAPER FEED 40 & 40 lines/inch

INITIAL RECORDER DELAY (thumbwheel) 0 & 4 sec

SEISMIC ROLL NUMBER(S) UNIB #3 & BA35 #2

2. MAGNETIC TAPE RECORDER

TYPE H-P 3968A - 8 TRK - 1/4 in

TAPE SPEED 3 3/4 in/sec

TAPE/TACH SERVO [✓ On] Off

FOOTAGE COUNTER: 1404 start run

3048 end run

CHANNEL	FM/DIR	INFORMATION
1	[✓ FM] DIR	vertical incidence seismic - Teledyne amp
2	[✓ FM] DIR	Right unfiltered sonobuoy seismic
3	[✓ FM] DIR	Left filtered sonobuoy seismic
4	[✓ FM] DIR	Right filtered sonobuoy seismi
5	[✓ FM] DIR	H-P precision drive frequency
6	[✓ FM] DIR	Waterbreak - right channel
7	~~FM~~ [✓ DIR]	Trigger
8	~~FM~~ [✓ DIR]	Voice & WWV

IV. SONOBUOY GEAR

1. SONOBUOY TYPE SB-76 SER. NO. 11971

CHANNEL 20L AMPS: [✓ Fixed gain] AGC

PHONE DEPTH 120 ft DELAY RELEASE 20 ~~sec~~ min

FREQUENCY BAND 76 MHz 171 MHz [✓ 74 MHz]

2. RECEIVER (Initial)

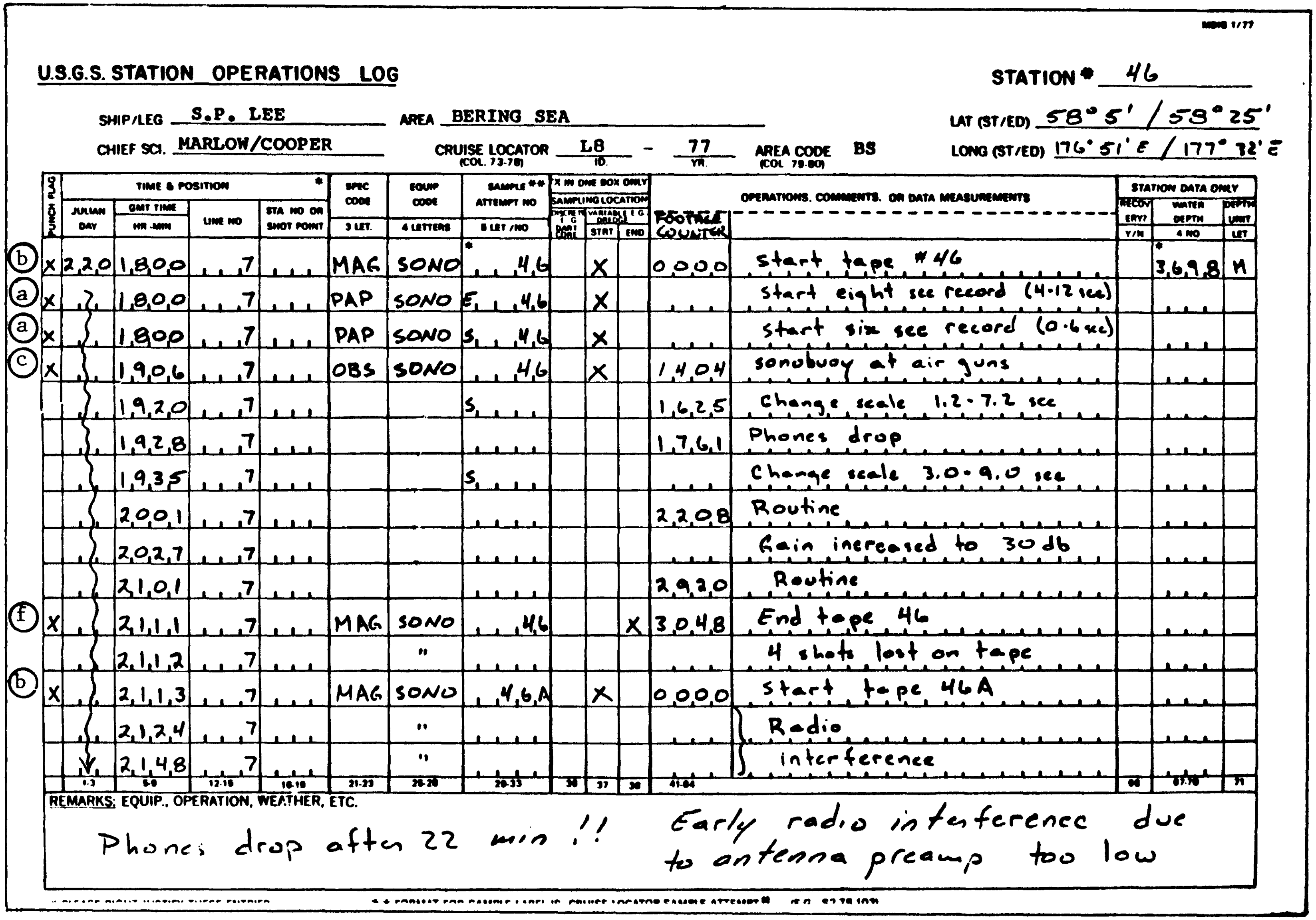
FRONT FILTERS HI OUT LOW 5

GAIN 12 dB

BACK 5 Hz LOCUT SWITCH [✓ Off] On

WATER BREAK ADD Off [✓ On]

FIGURE 18 EXAMPLE OF HOW TO FILL OUT SONOBUOY STATION LOG (SEE PAGE 51)



[illegible] 1/77

U.S.G.S. STATION OPERATIONS LOG

STATION # 46

SHIP/LEG S.P. LEE AREA BERING SEA

LAT (ST/ED) 58° 5' / 58° 25'

CHIEF SCI. MARLOW/COOPER CRUISE LOCATOR (COL. 73-78) L8 (ID.) - 77 (YR.) AREA CODE (COL 79-80) BS

LONG (ST/ED) 176° 51' E / 177° 32' E

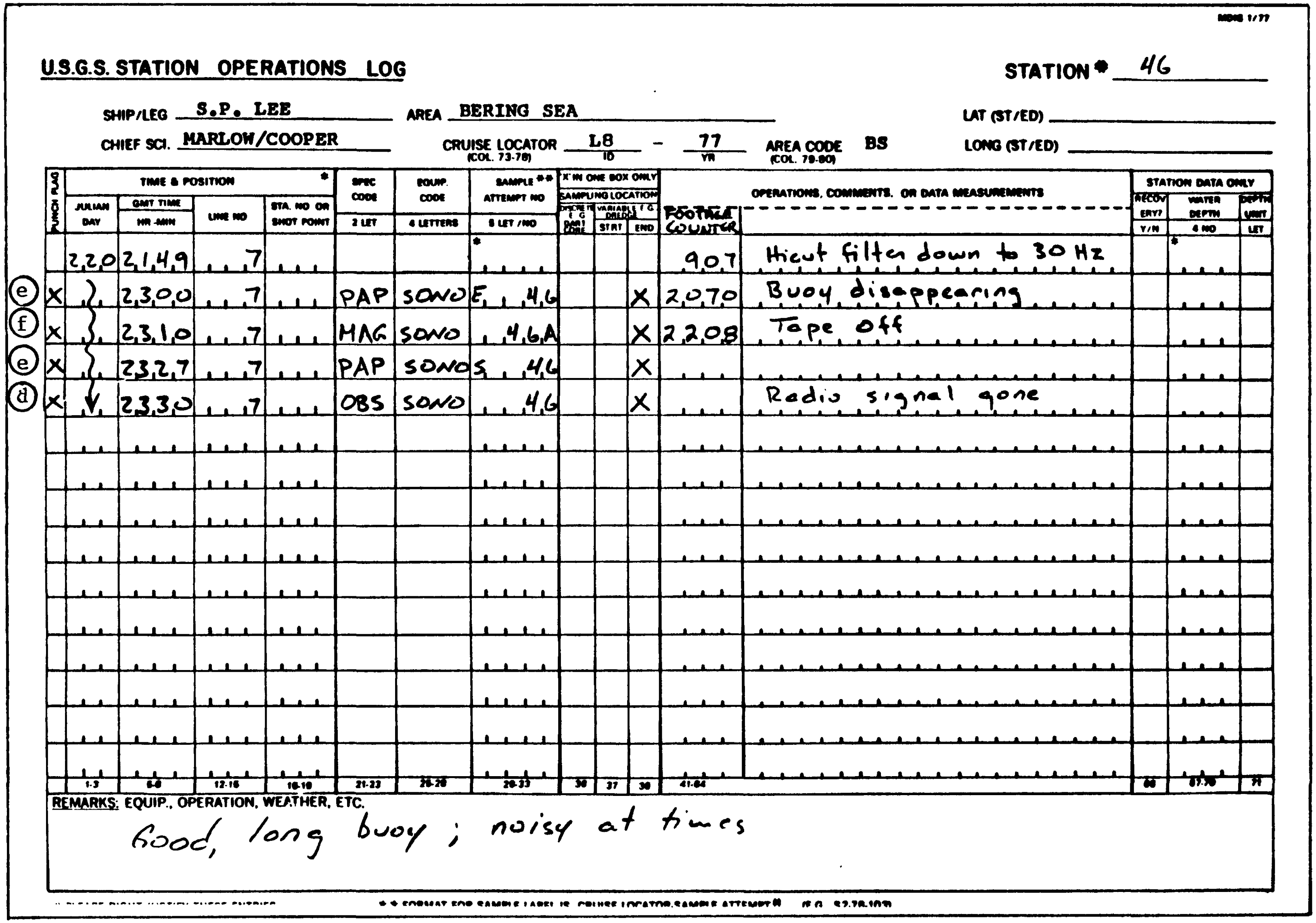
	PUNCH FLAG	TIME & POSITION *				SPEC CODE	EQUIP CODE	SAMPLE ** ATTEMPT NO	X IN ONE BOX ONLY SAMPLING LOCATION			OPERATIONS, COMMENTS, OR DATA MEASUREMENTS		STATION DATA ONLY		
		JULIAN DAY	GMT TIME HR-MIN	LINE NO	STA NO OR SHOT POINT	3 LET.	4 LETTERS	8 LET/NO	DISCRETE E.G. DART CORE	VARIABLE E.G. DREDGE STRT	END	FOOTAGE COUNTER		RECOVERY? Y/N	WATER DEPTH 4 NO	DEPTH UNIT LET
b	X	220	1800	7		MAG	SONO	* 46		X		0000	Start tape #46		* 3698	M
a	X		1800	7		PAP	SONO	E 46		X			Start eight sec record (4-12 sec)			
a	X		1800	7		PAP	SONO	S 46		X			start six sec record (0-6 sec)			
c	X		1906	7		OBS	SONO	46		X		1404	sonobuoy at air guns			
			1920	7				S				1625	Change scale 1.2-7.2 sec			
			1928	7								1761	Phones drop			
			1935	7				S					Change scale 3.0-9.0 sec			
			2001	7								2208	Routine			
			2027	7									Gain increased to 30db			
			2101	7								2920	Routine			
f	X		2111	7		MAG	SONO	46			X	3048	End tape 46			
			2112	7			"						4 shots lost on tape			
b	X		2113	7		MAG	SONO	46A		X		0000	Start tape 46A			
			2124	7			"						Radio			
			2148	7			"						interference			
		1-3	5-8	12-15	16-19	21-23	26-28	29-33	36	37	38	41-64		66	67-70	71

REMARKS: EQUIP., OPERATION, WEATHER, ETC.

Phones drop after 22 min !! Early radio interference due to antenna preamp too low

* PLEASE RIGHT JUSTIFY THESE ENTRIES ** FORMAT FOR SAMPLE LABEL ID, CRUISE LOCATOR SAMPLE ATTEMPT # (E.G. [illegible])

FIGURE 19 EXAMPLE OF HOW TO USE STATION OPERATIONS LOG TO RECORD SONOBUOY DATA (SEE PAGES 51 AND 52)



MDMS 1/77

U.S.G.S. STATION OPERATIONS LOG

STATION # 46

SHIP/LEG S.P. LEE AREA BERING SEA LAT (ST/ED) ______

CHIEF SCI. MARLOW/COOPER CRUISE LOCATOR (COL. 73-78) L8 (ID) - 77 (YR) AREA CODE (COL. 79-80) BS LONG (ST/ED) ______

	PUNCH FLAG	TIME & POSITION *				SPEC CODE	EQUIP. CODE	SAMPLE ** ATTEMPT NO	'X' IN ONE BOX ONLY SAMPLING LOCATION			FOOTAGE COUNTER	OPERATIONS, COMMENTS, OR DATA MEASUREMENTS	STATION DATA ONLY		
		JULIAN DAY	GMT TIME HR-MIN	LINE NO	STA. NO OR SHOT POINT	2 LET	4 LETTERS	5 LET/NO	DISCRETE E.G. DART CORE	VARIABLE E.G. DREDGE STRT	END			RECOVERY? Y/N	WATER DEPTH 4 NO	DEPTH UNIT LET
		220	2149	7				*				907	Hicut filter down to 30 Hz		*	
ⓔ	X	}	2300	7		PAP	SONO	E 46			X	2070	Buoy disappearing			
ⓕ	X	}	2310	7		MAG	SONO	46A			X	2208	Tape off			
ⓔ	X	}	2327	7		PAP	SONO	S 46			X					
ⓓ	X	↓	2330	7		OBS	SONO	46			X		Radio signal gone			
		1-3	6-8	12-15	16-19	21-23	26-28	29-33	36	37	38	41-64		66	67-70	71

REMARKS: EQUIP., OPERATION, WEATHER, ETC.

Good, long buoy ; noisy at times

* PLEASE RIGHT JUSTIFY THESE ENTRIES ** FORMAT FOR SAMPLE LABEL IS CRUISE LOCATOR-SAMPLE ATTEMPT# (E.G. S2-78-10S)

FIGURE 19 (CONTINUED)

5. If more than one buoy is being recorded on the same magnetic tape, then use the number of the first buoy deployed as the tape number. If buoys 3, 4, and 5 are being recorded, the tape should be number 3.

6. Other sequential entries are needed also. The general rule is that whenever you 'twiddle a knob', make an entry on this logsheet. Items that normally happen and should be noted, include:

   a. Amplifier gain and filter changes

   b. Recorder delay changes

   c. Equipment failures

   d. Event marks

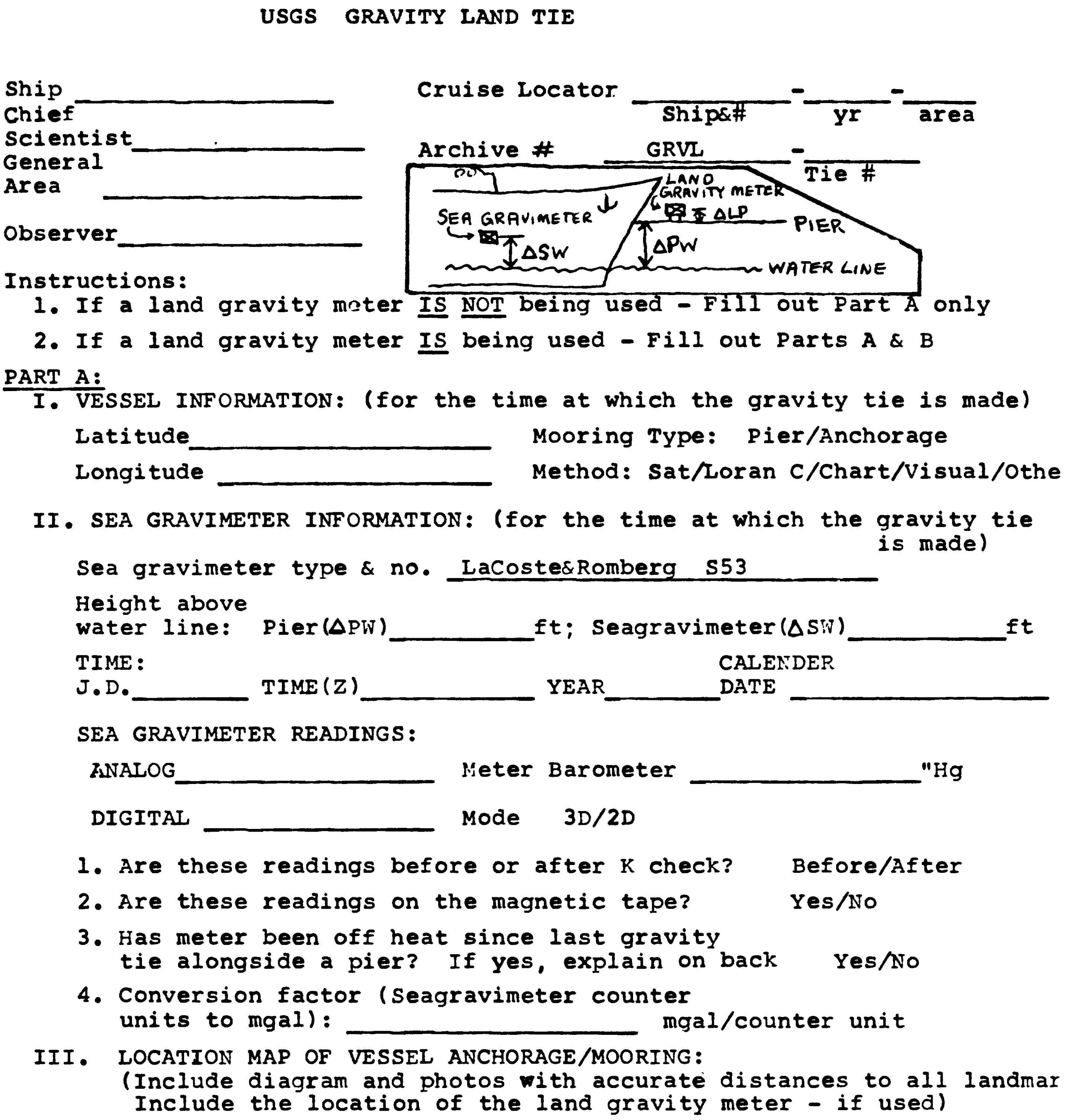
   e. Magnetic tape changes

III. U.S.G.S. GRAVITY LAND TIE - Gravity land ties are an essential part of a marine gravity survey since they provide information on the 'drift' of the gravity meter during the survey. Land ties must be made at the start and end of every cruise to ensure the accuracy of the gravity measurements. Reliable gravity land ties can only be made when the ship is at a quiet dockside location. Anchorage sites are unusable for two reasons: excessive ship motion and inaccurate absolute gravity estimates. The Gravity Land Tie form is in two parts (pages 58,59). Part A provides information about the sea gravimeter and the dockside location. Part B outlines the steps that are necessary to 'transfer' (via land gravity meter) an absolute gravity value from a land base station to a dockside location alongside the ship. There are two essential requirements in filling out the Land Tie form:

1. Fill out Part A of the form every time the ship comes into port, (a few hours after docking) and again when the ship leaves port (a few hours prior to departure). 'Tears' in the gravity data can occur at any time.

2. Take a polaroid photograph of the ship's position alongside the dock and tape the photo to the form. An accurate map will suffice if exact footage measurements are given.

Remember, gravity data are the 2nd most expensive data (below multi-channel seismic) collected by the Office of Marine Geology. Protect this investment by ensuring high quality data. Always fill out the Gravity Land Tie forms.



USGS GRAVITY LAND TIE

Ship ____________________ Cruise Locator ____________-______-_____
Chief   Ship&#   yr   area
Scientist____________________
General   Archive #   GRVL   -________
Area ____________________   Tie #

LAND
GRAVITY METER
SEA GRAVIMETER
ΔSW
ALP
ΔPW
PIER
WATER LINE

Observer____________________

Instructions:

1. If a land gravity meter IS NOT being used - Fill out Part A only

2. If a land gravity meter IS being used - Fill out Parts A & B

PART A:

I. VESSEL INFORMATION: (for the time at which the gravity tie is made)

Latitude____________________ Mooring Type: Pier/Anchorage

Longitude ____________________ Method: Sat/Loran C/Chart/Visual/Othe

II. SEA GRAVIMETER INFORMATION: (for the time at which the gravity tie is made)

Sea gravimeter type & no. LaCoste&Romberg S53

Height above
water line: Pier(ΔPW)__________ft; Seagravimeter(ΔSW)___________ft

TIME:
J.D.________ TIME(Z)____________ YEAR________ CALENDER DATE ______________

SEA GRAVIMETER READINGS:

ANALOG__________________ Meter Barometer ________________"Hg

DIGITAL ________________ Mode 3D/2D

1. Are these readings before or after K check? Before/After

2. Are these readings on the magnetic tape? Yes/No

3. Has meter been off heat since last gravity tie alongside a pier? If yes, explain on back Yes/No

4. Conversion factor (Seagravimeter counter units to mgal): ____________________ mgal/counter unit

III. LOCATION MAP OF VESSEL ANCHORAGE/MOORING:
(Include diagram and photos with accurate distances to all landmar
Include the location of the land gravity meter - if used)

FIGURE 20 EXAMPLE OF U.S.G.S. GRAVITY LAND TIE FORM (SEE PAGES 56 & 57)

USGS GRAVITY LAND TIE (Cont)

Locator __________-________-________
Ship & # Yr Area

PART B:

IV. LAND GRAVITY INFORMATION:

Land gravity meter type & no.: ______________________________

Gravity Base name: ______________________________

General location: ______________________________ (Show map below)

Absolute gravity: ______________________________ Gals

Bench mark #: ______________ Reference: ______________

V. BASE GRAVITY TO DOCK GRAVITY: (3 loop tie)

Loop #	Place	Time	Gravity meter counter readings	Height of land meter above water line
1	Dock			ft
	Base			
2	Dock			ft
	Base			
3	Dock			ft
	Base			
	Dock			ft

1. Height of land meter above pier(ΔLP): ______________ ft

2. Conversion factor (land gravity meter counter units to mgal): ______________ mgal/counter uni

VI. LOCATION MAP OF GRAVITY BASE STATION:
(Include diagram and photos with accurate distances to all landmark

FIGURE 20 (CONTINUED)

SECTION FIVE

This section contains lists of the Media/Specification and Data Codes which are used to initially identify data in ship-board logs as it is collected at sea. Following these lists, are lists and diagrams of the "ladder codes" which are assigned to the data by the computer.

Media/Specification Codes consist of 3-letter mnemonic codes (e.g., PAP for paper) used to identify either the media types on which data are recorded or a very general grouping of the data type (e.g., SAM for samples). Used in this way, Media/Specification Codes correspond to the 2nd identification level of the computer-generated "ladder codes". Data Codes are made up of 4 characters (letters or integers) which mnemonically identify specific types of data or data systems (e.g., SCAN for side scan sonar). They have the same function as the 4th identification level of a "ladder code". A piece of data is uniquely identified when the proper combination of Media/Specification and Data Code is assigned to it.

"Ladder codes" are non-mnemonic groups of four characters (letters or integers) which are assigned to a piece of data by the computer. These codes describe a 4-level hierarchal organization for oceanographic data where each character of the code represents one of the identification levels. The first level of a "ladder code", represented by the left-most character of the code, specifies which of four general categories of data a piece of data belongs to, namely: geologic, geophysical, hydrologic, or vessel. The next lower level is generally used to identify the media type that the data are recorded on (same function as a Media/Specification Code). The third level is used to identify equipment systems. The fourth and lowest level identifies specific pieces of equipment or sampling devices (same function as a Data Code). "Ladder codes" are used primarily to give a user more selectivity in picking various types and combinations of data from a general index. An example of a "ladder code" might be: XAHS; where the 'X' stands for geophysical data; the 'A' stands for analog paper roll; the 'H' stands for high resolution seismic; and the 'S' stands for side scan sonar.

MEDIA/SPECIFICATION CODES FOR ALL DATA TYPES

NAME	MEDIA CODE
I. RECORDING MEDIA	
PAPER ROLLS	PAP
MAGNETIC TAPE (DIGITAL)	MAG
ANALOG/VIDEO MAGNETIC TAPE	AVM
PHOTOGRAPH	PHO
PRINTER LISTING (OR PRINT TAPE)	PRN
PUNCH TAPE	PUN
MAPS AND CHARTS	MAP
COMPUTER CARDS	CRD
II. SPECIFICATIONS	SPEC. CODE
SAMPLING ATTEMPTS	SAM
NAVIGATION (SHIP MOVEMENTS)	NAV
TRACKLINES	"
STATIONS	"
*PORT STOPS	"
*CRUISE DATES	"
*INVENTORY LISTS	
PERSONNEL ON BOARD	PER
EQUIPMENT SYSTEMS ON USED	EQP
OBSERVATIONS	
NUMERICAL	OBS
MISCELLANEOUS	OBS
DEPLOYMENTS (SONOBUOYS, EXPLOSIVES, ON-BOTTOM SEISMOMETERS, SCUBA DIVES, ETC.)	DEP
EQUIPMENT STATUS (MULTI-INSTRUMENT AND NAVIGATIONAL SYSTEMS DATA ONLY)	SYS
NAVIGATION	"
GRAVITY	"
MAGNETICS	"
DIGITRAK	"
ETC.	"

* TO BE HANDLED BY CRUISE CURATOR

DATA/EQUIPMENT CODES

GEOLOGIC DATA

NAME	DATA(EQUIPMENT) CODE
I. CORING DEVICES	
DART CORE	DART
PISTON CORE	PSTN
VIBRATING CORE	VIBR
BOX CORE (CROPPED FROM SHIP)	BOXS
BOX CORE (HAND HELD)	BOXH
GRAVITY CORE	GRAV
DRILL CORE	DRIL
BOOMERANG CORE	RANG
ALPINE CORE	ALPN
PHLEGER CORE	FLAG
HYDROPLASTIC CORE	HYPC
II. DREDGING DEVICES	
CHAIN DREDGE	CHAN
PIPE DREDGE	PIPE
III. SEAFLOOR GRABBING DEVICES	
FREE FALL GRAB	FREE
SHIPEK GRAB	SHIP
VAN VEEN GRAB	VANV
PETERSON GRAB	PETR
PONAR GRAB	PONR
CAMPBELL GRAB	CAMP
ORANGE PEEL GRAB	PEEL
HAND GRAB	HAND
SOUTAR GRAB	SUTR
DIETZ/LAFOND GRAB	DETZ
IV. MISCELLANEOUS DATA	
LAND SAMPLE	LAND
SEAFLOOR CAMERA	CMRA
TELEVISION	TVIS
SUSPENDED SEDIMENT	SUSP

DATA/EQUIPMENT CODES

GEOLOGIC DATA (CONT)

NAME	DATA(EQUIPMENT) CODE
UNDERWAY SAMPLER	UWAY
SCUBA DIVE	SDIV
PENETROMETER	PENT
SIDESCAN MARKER	MARK

DATA/EQUIPMENT CODES

GEOPHYSICAL DATA

	NAME	DATA(EQUIPMENT) CODE
I.	SEISMIC REFLECTION	
	MULTICHANNEL (24)	MC24
	MULTICHANNEL (36)	MC36
	MULTICHANNEL (48)	MC48
	MULTICHANNEL (72)	MC72
	SINGLE CHANNEL SEISMIC (AIR GUNS)	SCAG
	SINGLE CHANNEL SEISMIC (ARCER)	SCAR
	SINGLE CHANNEL SEISMIC (AIR + ARCER)	SCAA
	SINGLE CHANNEL SEISMIC (AIRGUN MULTICHANNEL MONITOR)	SCAM
	BURG SEISMIC (AIR GUNS)	BRGA
	BURG SEISMIC (SPARKER)	BRGS
	BURG SEISMIC (AIRGUNS AND SPARKER)	BRG2
II.	HIGH RESOLUTION SEISMIC	
	UNIBOOM	UNIB
	MINI SPARKER	MSPK
	DEL NORTE	DELN
	SIDE SCAN SONAR	SIDE
	UNIBOOM/MINISPARKER COMBINED	UBMS
III.	BATHYMETRY	
	3.5 KHZ BATHYMETRY	BA35
	7 KHZ BATHYMETRY	BA7K
	12 KHZ BATHYMETRY	BA12
	48 KHZ BATHYMETRY	BA48
	200 KHZ BATHYMETRY	BA2H
	200 KHZ + 7 KHZ BATHYMETRY	BATS
	BATHYMETRY WITH NAVIGATION	BATN
IV.	GRAVITY	
	GRAVITY (SHIPBOARD)	GRVS
	GRAVITY (LAND TIE)	GRVL
	GRAVITY WITH BATHY	GRVB
	GRAVITY WITH MAGNETICS	GRVM

DATA/EQUIPMENT CODES

GEOPHYSICAL DATA (CONT)

NAME	DATA(EQUIPMENT) CODE
IV. GRAVITY (CONT)	
GRAVITY WITH BATHY & MAGNETICS	GRV2
GRAVITY WITH BATHY, MAGNETICS & NAVIGATION	GRV3
GRAVITY WITH BATHY & NAVIGATION	GRBN
V. MAGNETICS	
MAGNETOMETER (SHIPBOARD)	MAGS
MAGNETOMETER (LAND STATION)	MAGL
GRADIOMETER	GRAD
MAGNETICS WITH BATHY	MAGB
MAGNETICS WITH NAVIGATION	MAGN
MAGNETICS WITH BATHY & NAVIGATION	MABN
VI. OTHER GEOPHYSICAL EQUIPMENT	
SONOBUOY	SONO
AIR GUN SYSTEM	AGUN
SPARKER SYSTEM	SPRK
DIGITRAK (DIGITAL BATHY)*	DIGT
EXPLOSIVE CHARGE	BOOM
OCEAN BOTTOM SEISMOMETER	BOTS
HEAT FLOW GAUGE	HEAT

* USED WITH MULT-INSTRUMENT SYSTEMS AND WITH SPEC.CODE = "SYS"

DATA/EQUIPMENT CODES

HYDROLOGIC DATA

NAME	DATA(EQUIPMENT) CODE
I. WATER SAMPLING DEVICES	
NANSEN BOTTLE	NANS
NISKIN BOTTLE	NISK
VAN DORN BOTTLE	VAND
WATER BUCKET	WATB
II. TURBIDITY/TEMPERATURE/SALINITY METERS	
TRANSMISSOMETER	TRAN
NEPHELOMETER	NEPH
SECCHI DISK	SECC
THERMO-SALINITY METER	TSAL
EXPENDABLE BATHY-THERMO GRAPH	XBTG
TEMPERATURE GAUGE	TEMP
III. CURRENT METERS	
AANDERAA CURRENT METER	ANDR
GEODYNE CURRENT METER	GEOD
BENDIX CURRENT METER	BEND
MARSH McBIRNEY CURRENT METER	MARS
HYDROPRODUCTS CURRENT METER	HYDP
IV. INSTRUMENT PACKAGES	
BECKMAN STC	BECK
CTD METER	CTDM
GEOPROBE	GEOP
V. TIDE GAUGES OR METERS	ATYD
VI. MISCELLANEOUS	
WAVE PRESSURE SENSOR	WAVE
PLANKTON NET	PLAN
RIPPLE PROFILER	RIPL

DATA/EQUIPMENT CODES

MISCELLANEOUS DATA

NAME	DATA(EQUIPMENT) CODE
I. SCIENTIFIC PERSONNEL	
CHIEF SCIENTIST	KING
CRUISE CURATOR	DAFE
GEOLOGIST	GEOL
GEOPHYSICIST	GEOP
GEOCHEMIST	GEOC
BIOLOGIST	BIOL
OCEANOGRAPHER	OCEO
ELECTRONIC TECHNICIAN	ETEC
MECHANICAL TECHNICIAN	MTEC
WATCHSTANDER	WACH
NAVIGATOR	NAVG
UNSPECIFIED PRINCIPAL INVESTIGATOR	UNSP
II. SHIP'S PERSONNEL	
CAPTAIN	CAPT
CHIEF ENGINEER	CENG
CHIEF MATE	MATE

DATA/EQUIPMENT CODES

VESSEL (NAVIGATION) DATA

NAME	DATA (EQUIPMENT) CODE
I. SHIP MOVEMENTS	
TRACKLINE TIMES	TRAK
STATION TIMES (WHERE MULTIPLE SAMPLES TAKEN)	STAT
PORT STOPS	PORT
CRUISE DATES (START AND END)	CRUZ
II. NAVIGATION EQUIPMENT	
SATELLITE NAVIGATION (SAT-NAV)	SATN
DOPPLER SONAR	DOPP
LORAN A	LORA
LORAN C	LORC
LORAN C RHO-RHO	LORR
RADAR	RADR
MINIRANGER	MRAN
DELNORTE (RANGE-RANGE)	NORT
HIFIX	HFIX
RAYDIST	RDIS
SHIPBOARD INTEGRATED NAVIGATION SYSTEM	SINS
COURSE RECORDER	CREC
LORAC (BEAT FREQUENCY SYSTEM)	LORK
III. MISCELLANEOUS DATA	
DEAD RECKONING NAVIGATION PLOT (FROM BRIDGE)	DRBR
DEAD RECKONING NAV PLOT (FROM SCIENTIFIC LAB)	DRSL
DEAD RECKONING NAV PLOT (FROM COMPUTER)	DRCP

LADDER CODE CROSS-REFERENCE LIST

SCIENTIFIC CRUISE/MEDIA TYPE

	LEVEL I	LEVEL II
A		Paper Rolls
B		
C		Computer Cards
D		Deployments (Sonobuoys, Explosives, etc)
E		
F		Paper Punch Tapes
G	Geologic Data	Paper Printer Listing
H	Hydrographic Data	Maps and Plots
I		
J		
K		Ship Movements
L		
M		Digital Magnetic Tape
N		
O		Numerical Data Observation
P		Photo Observations
Q		
R		Personnel
S		Physical Sample
T		Analog Magnetic Tapes
U		
V	Vessel Data	
W		Inventory of Scientific Gear
X	Geophysical Data	Operation of Scientific Gear
Y		
Z		

GEOLOGIC DATA

	LEVEL III	LEVEL IV
A		Alpine Corer
B		Box Corer
C	Seafloor Corer	Chain Dredge
D	Seafloor Dredge	Dart Corer
E	Engineering Properties	Dietz-LaFond Grab
F		*Freefall Grab / Phleger Corer
G		Gravity Corer
H		Penetrometer
I		Television
J	Underwater Dive	Sea Floor Camera
K	Seafloor Grab	Shipek Grab
L	Land Grab	Hydroplastic Corer
M	Miscellaneous	*Handsample, Marker
N		Box Corer (hand held)
O	Suspended Sediment	Orange Peel Grab
P		Piston Corer
Q	Underwater Camera	Scuba Dive
R		Vibrating Corer
S		Boomerang Corer
T		Drill Corer
U	Seafloor Underway Sampler	Soutar Grab
V		Van Veen Grab
W		Pipe Dredge
X		Peterson Grab
Y		Ponar Grab
Z		Campbell Grab

* Two types of equipment per letter.

GEOPHYSICAL DATA

	LEVEL III	LEVEL IV
A		Airgun
B	Bathymetry (3.5 - 200 KHZ)	Gravity with Bathymetry & Navigation
C		Heat Gauge
D		Del Norte
E		Explosives
F	Heat Flow	Sonobuoy
G	Shipboard Gravity	Gravity
H	High Resolution Acoustic (150-3499 HZ)	48 Channel Record
I		Digitrack
J	Station Gravity	
K	Station Magnetics	36 Channel Record
L		Magnetics with Bathymetry & Navigation
M	Shipboard Magnetics	*Magnetics / Multichannel Monitor
N		*Land Station / 48 KHZ Transducer
O		Ocean Bottom Seismometer
P		72 Channel Record
Q	Burg Seismic	
R		24 Channel Record
S		Side Scan Sonar
T	Single Channel Seismic (150 HZ)	*Land Tie / 200 KHZ Transducer
U		Uniboom
V		Magnetics with Navigation
W	Wide Angle Reflection/Refraction	12 KHZ Transducer
X	Multichannel Seismic	*3.5 KHZ Transducer / Sparker
Y		7 KHZ Transducer
Z	Gradiometer	*Gradiometer / Minisparker

* Two types of equipment per letter.

GEOPHYSICAL DATA (CONT)

LEVEL III	LEVEL IV
1	Gravity and Magnetics
2	Gravity and Bathymetry
3	Gravity, Magnetics and Bathymetry
4	Magnetics and Bathymetry
5	Airgun and Sparker (combined fire)
6	Gravity, Bathymetry, Navigation, Magnetics
7	Uniboom and Minisparker
8	200 + 7 KHZ Bathymetry
9	Bathymetry with Navigation

HYDROLOGIC DATA

	LEVEL III	LEVEL IV
A		Aanderaa Tide Gauge
B		Beckman STC
C	Current Meter (Speed/Direction)	CTD Meter
D		Geodyne Current Meter
E		Expendable Bathy Thermograph
F	Water Filtering Device	Ripple Profiler
G		Geoprobe
H		Hydroproducts Current Meter
I	Instrument Package	Secchi Disk
J		
K	Temperature/Salinity/Conductivity Gauges	Nansen Bottle
L	Tide Gauge (Lunar)	Plankton Net
M	Miscellaneous	Marsh-McBirney Current Meter
N	Net	Nisken Bottle
O		Suspended Sediment
P	Pressure Sensor	Nephelometer
Q		Thermo-Salinity Meter
R		Aanderaa Current Meter
S	Sonar	Wave Pressure Sensor
T	Turbidity	Transmissometer (Hydroproducts)
U		
V		Van Dorn Bottle
W	Water Sampler	Water Bucket
X		Bendix Current Meter
Y		
Z		Temperature Gauge

VESSEL DATA

	LEVEL III	LEVEL IV	
		PERSONNEL	NAVIGATION
A			Loran A
B		Biologist	Bridge DR Plot
C		Captain	Loran C
D		Cruise Curator	Doppler Sonar
E		Chief Engineer	
F			Course Recorder
G		Geologist	
H			S/E of Cruise
I			
J			
K			Discrete Station Times
L			Lorac (beat freq. system)
M		Chief Mate	Mini-Ranger (Motorola)
N	Navigation	Navigator	
O		Oceanographer	
P			Port Stops
Q		Electronic Tech.	Loran C (Rho Rho)
R	Scientific Staff		Radar
S		Chief Scientist	Satellite Navigation
T		Mechanical Tech.	Trackline Times
U		Unspecified Principal Investigator	
V			Delnorte Range-Range
W		Watchstander	Computer DR Plot
X		Geophysicist	Hi-Fix
Y	Ship's Crew	Geochemist	Raydist
Z			Scientific Lab DR plot
			6 Shipboard Integrated Navigation System (SIN SYSTEM)

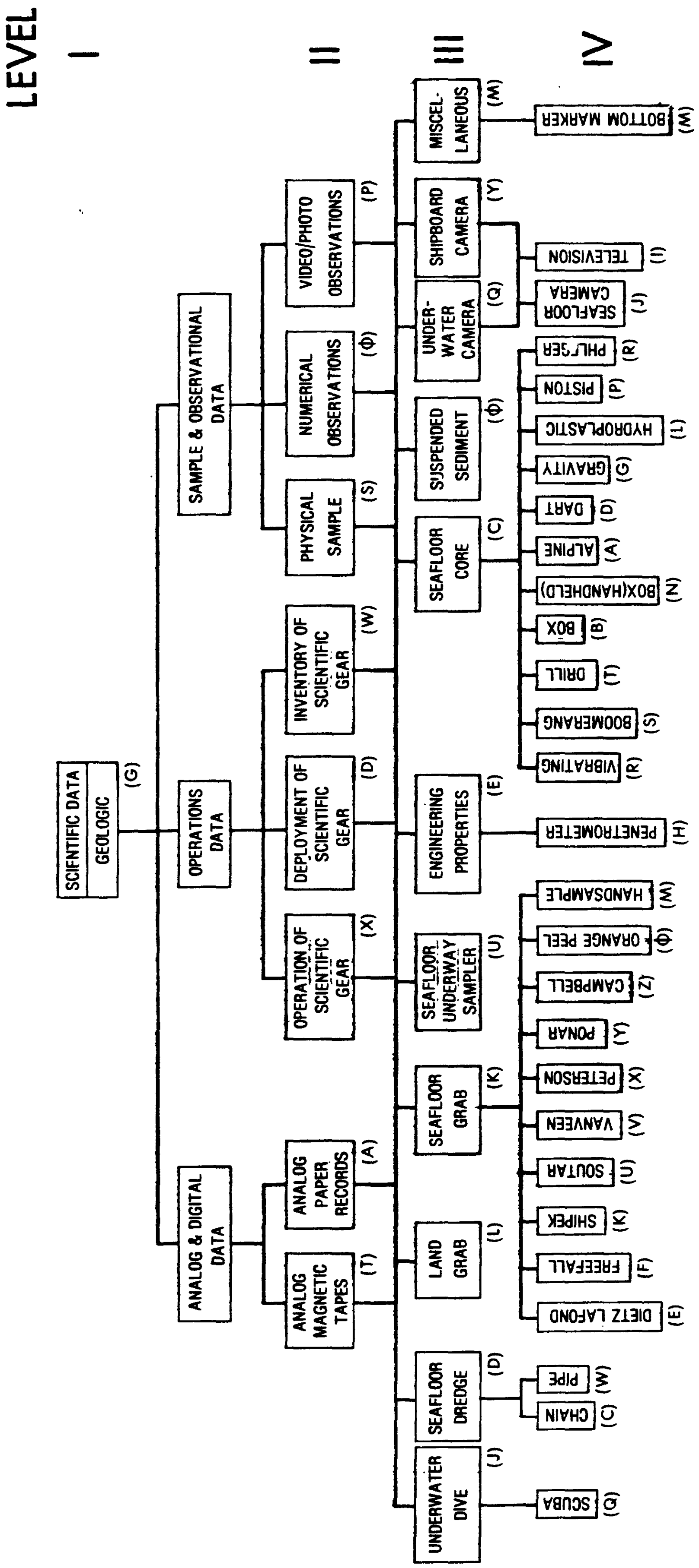


FIGURE 21 LADDER CODE DIAGRAM FOR GEOLOGIC DATA

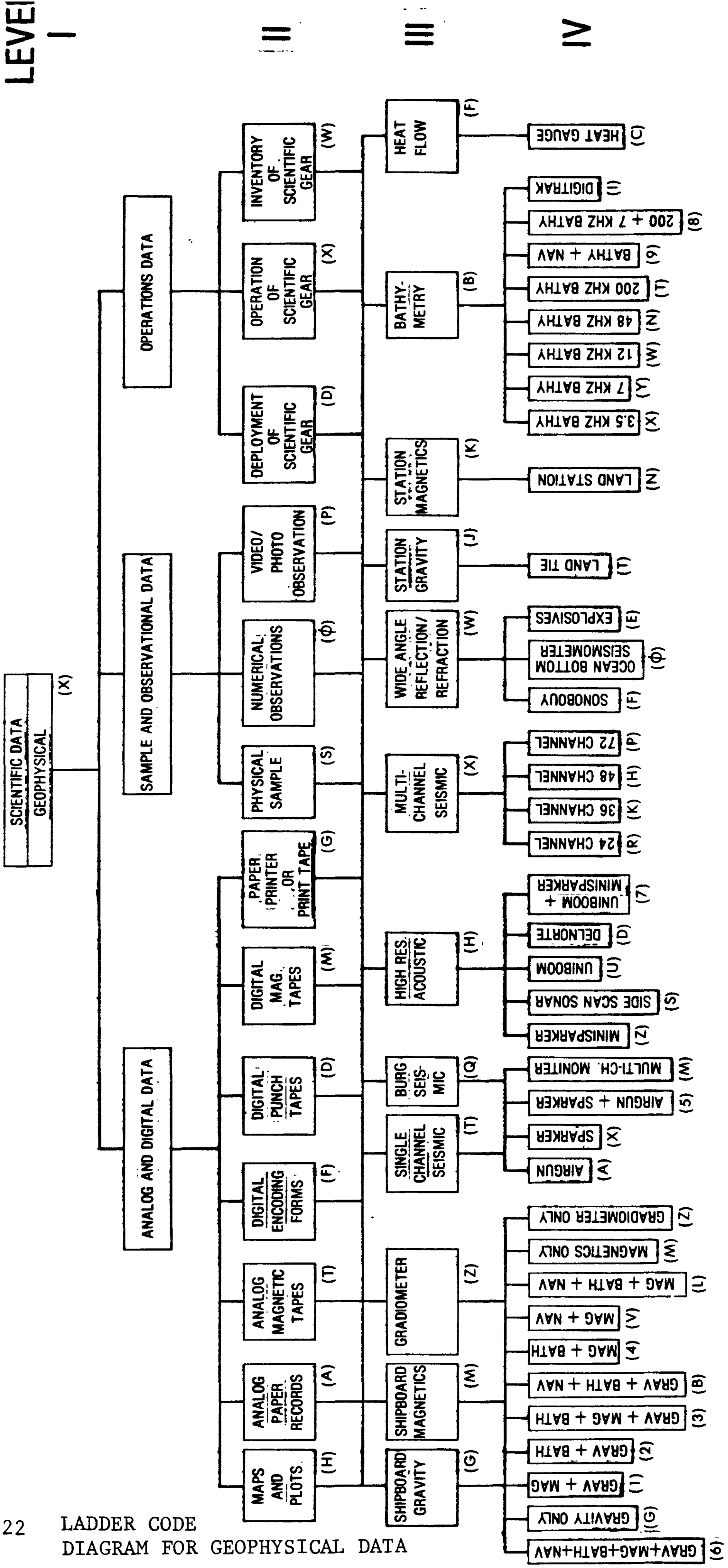


FIGURE 22 LADDER CODE DIAGRAM FOR GEOPHYSICAL DATA

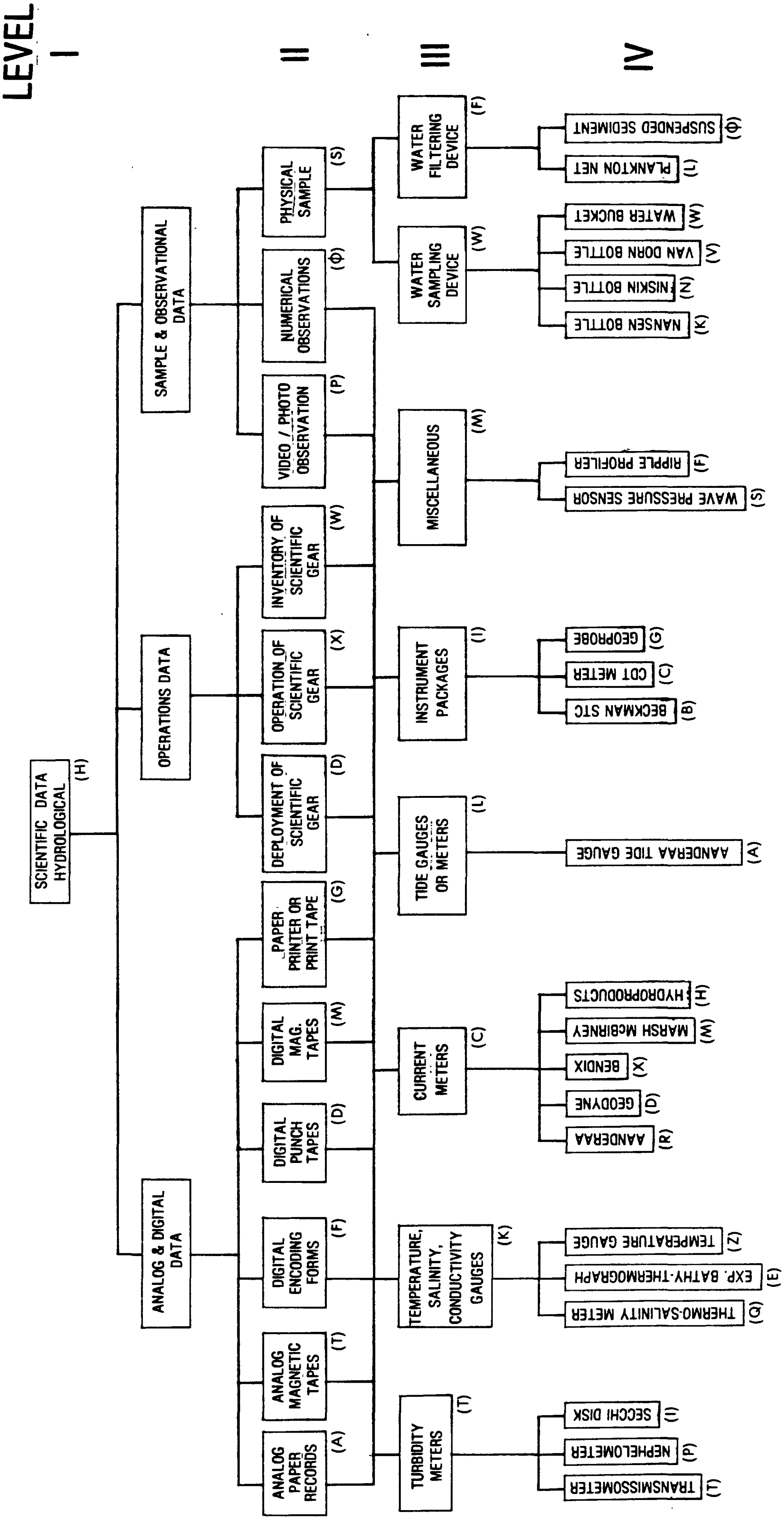


FIGURE 23 LADDER CODE DIAGRAM FOR HYDROLOGICAL DATA

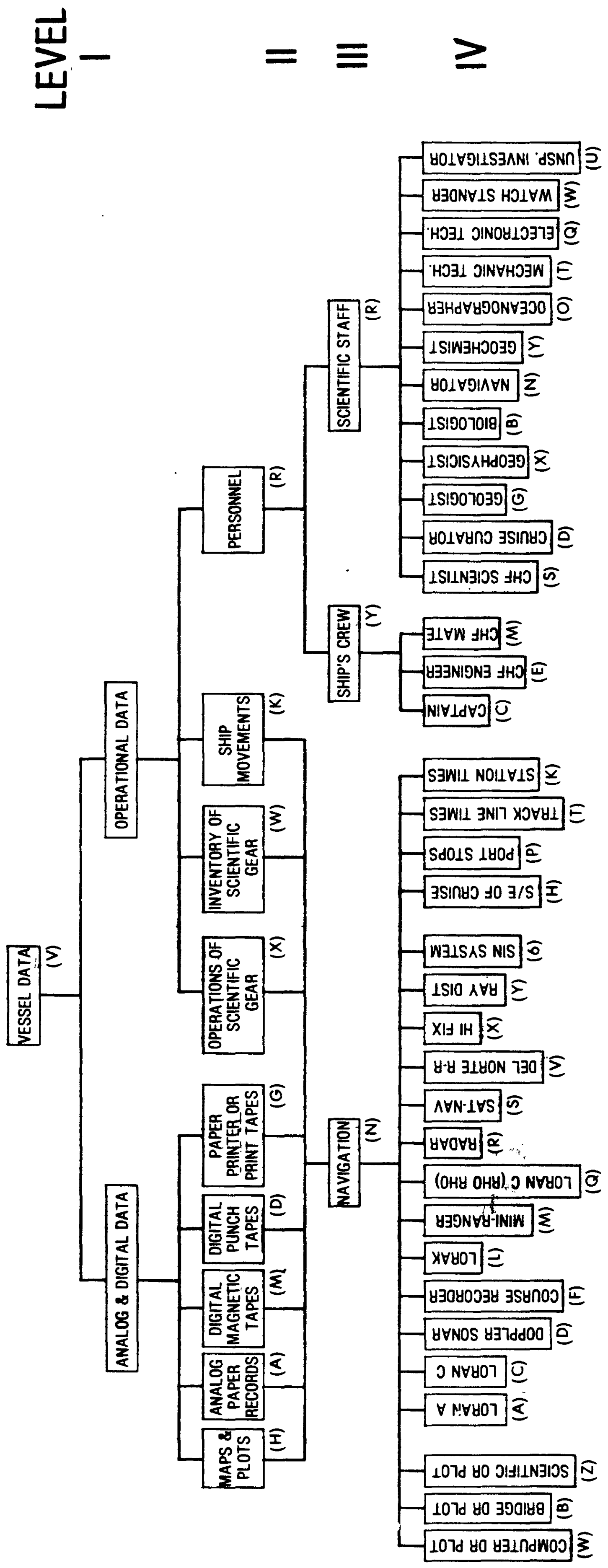


FIGURE 24 LADDER CODE DIAGRAM FOR VESSEL DATA

SECTION SIX

This section contains the computer programs and documentation for the computer programs that are used to manipulate Cruise Data Index files. These programs are EDITCDI, STATCDI, and PLOTCDI.

EDITCDI was created to sort, edit, organize and display in a readable format Cruise Data Indices. STATCDI was written to generate cruise statistics from the indices. PLOTCDI is used to select navigation positions for the times data were being recorded by a particular data system and is used in conjunction with Paul Celluzzi's plot program NAVPLT to plot tracklines for each data system.

Documentation and output examples have been included to enable a user to run the programs with little or no assistance and also to be able to understand the basic structure of each program. Though these programs have been set up to run on the Honeywell 68 series 80 computer, it would not take too much work to convert them for use with other fortran compilers and computing systems.

Examples of the output for these programs can be found on the following pages:

EDITCDI OUTPUT

Cruise Data Index listing page 100

Cruise Data Index listing in 'readable format' page 101

STATCDI OUTPUT

'Cruise Report' based on a Cruise Data Index and navigational fixes for the cruise pages 158 - 160

PLOTCDI OUTPUT

Computer generated plot of 'Uniboom' trackline data from three separate cruises page 203

PROGRAM EDITCDI

```
PROGRAM EDITCDI - EDIT CRUISE DATA INDEX
  APPLICATION   - SPECIFIC TO CRUISE DATA INDEX
  FUNCTION      - ORGANIZATION AND DISPLAY OF CRUISE INDICES
  MODE          - INTERACTIVE


==========================================================================
                         I D E N T I F I C A T I O N
==========================================================================

  PROGRAM.......EDITCDI

  DESCRIPTION...THIS PROGRAM MANIPULATES A CRUISE DATA INDEX INPUT
                FILE FOR EDITING AND DISPLAY PURPOSES.  POSSIBLE
                MANIPULATIONS ON CDI FILES INCLUDE:
                (1) ORGANIZE IN CHRONOLOGICAL ORDER
                (2) ORGANIZE BY FIXED GROUPS OF DATA TYPES (HIERARCHAL
                    SORT)
                (3) MERGE NAVIGATION WITH EACH CDI RECORD BASED ON
                    TIME.
                (4) AUTOMATICALLY SORT INTO GROUPS OF MEDIA CODES
                    WHICH IN TURN ARE ORGANIZED IN GROUPS OF DATA
                    TYPES.
                (5) SELECTION OF SUB-DATA SETS CONSISTING OF RECORDS
                    OF A SINGLE SELECTION FACTOR OR A COMBINATION OF
                    SELECTION FACTORS.
                (6) DISPLAY IN AN EXPANDED FORMAT AMENABLE TO EASY
                    READING.
                (7) CONDENSE TO AN 80-COLUMN FORMAT FOR EFFICIENT
                    STORAGE ON CARDS, IF DESIRED.

  VERSION.......28 SEP 1977

  CONTACT.......TOM CHASE, BRADLEY LARSEN
                MARINE GEOLOGY, U.S. GEOLOGICAL SURVEY, MENLO PARK

  PROGRAMMING...B. LARSEN  ORIGINAL                             MAR 1976
                    (GENERAL STRUCTURE OF CDIMAIN BASED ON 1ST
                    GENERATION PROGRAM BY G. MCHENDRIE )
                    CDIMAIN, AUTOSR,  THSORT
                B. LARSEN  MODIFICATION                         APR 1976
                    ADD SELSCH, BLOCK_DATA
                B. LARSEN  MODIFICATION                         JUN 1976
                    ADD FINPRN, FINPUN
                B. LARSEN  MODIFICATION                         JUL 1976
                    ADD NAVMER (ORIGINAL BY G. MCHENDRIE)
                B. LARSEN  CONVERSION                           JAN 1977
                    CONVERTED FROM IBM 370 TO HONEYWELL MULTICS
```

```
PROGRAM EDITCDI


                B. LARSEN  MODIFICATION                        FEB 1977
                    MODIFIED INPUT FORMATS AND PORTIONS OF PROGRAMS
                    DEALING WITH SAMPLE DATA.
                B. LARSEN  MODIFICATION                        MAR 1977
                    ADD TITLE

  COMPUTERS.....HONEYWELL SERIES 60 UNDER MULTICS
  PROCESSORS....HONEYWELL MULTICS FORTRAN (RELEASE 4)

  INPUT.........(1) INTERACTIVE COMMANDS AND ANSWERS TO PROGRAM
                    PROMPTS (TERMINAL)
                (2) CARD IMAGE CDI LOG ENTRIES WITH SECONDS. (UNIT 5)
                (3) EXPANDED CDI LOG ENTRIES WITH DECIMAL MINUTES;
                    WITH DECIMAL MINUTES AND 'ADDITIONAL VARIABLES';
                    WITH DECIMAL MINUTES, 'ADDITIONAL VARIABLES' AND
                    NAVIGATION.  (UNIT 5)
                (4) CONDENSED CDI LOG ENTRIES (INHOUSE PUNCH) (UNIT 5)
                (5) NAVIGATION DATA
                    FORMATTED CARD IMAGE (UNIT 10), UNFORMATTED FROM
                    MAGNAVOX TAPE (UNIT 10)

  OUTPUT........(1) CDI RECORD IN SAME FORMAT AS INPUT (3) AND (4)
                (2) FINAL PRINT EXPANDED FORMAT
                (3) FINAL PUNCH CONDENSED 80 COLUMN CARD IMAGE
                    EITHER INHOUSE OR NGSDC FORMATS

  SUBROUTINES...(A) CDIMAIN  INTERACTIVE PROGRAM INPUT AND CONTROL
                (B) INCDI    INPUTS CRUISE DATA INDEX
                (C) AUTOSR   AUTOMATICALLY SORTS CDI RECORDS
                (D) THSORT   SORTS BY TIME OR HIERARCHAL NUMBER
                (E) SELSCH   SELECTS SUB-DATA SETS USING 'S'
                (F) NAVMER   MERGES NAVIGATION
                (G) FINPRN   EXPANDS CDI RECORDS INTO MORE READABLE
                             OUTPUT
                (H) FINPUN   CONDENSES CDI INTO 80 COLUMN RECORDS WITH
                             LAT/LONG
                (I) TITLE    ADDS A TITLE TO FINAL PRINT
                (J) BLOCK_DATA  STORES LABELED COMMON AND DATA
                             STATEMENTS

  EXECUTION.....PROGRAM CAN BE RUN BY INVOKING: bdafe$cdimain
                1ST TIME USERS WORKING UNDER A DIFFERENT USER ID,
                    ENTER: link >udd>Marine>BLarsen>bdafe
                TO LINK WITH THE SOURCE CODE: link
                    >udd>Marine>BLarsen>cdi.source.archive
                TO LINK WITH 'prefio': link
                    >udd>Marine>GMchendrie>maps>prefio
```

O P E R A T I O N

EXAMPLE TYPE

The following example is given to show how to initiate and direct an EDITCDI program run on a CDI input data set. The CDI data set to the example is assumed to have been freshly keypunched from cruise logs and to be error-free. The following manipulations will be performed:

(1) Input (minutes-seconds changed to decimal minutes).
(2) Automatic sort (routed through time-sort since it was not input in chronological order).
(3) Time sort again (AUTOSR disrupted the chronological order).
(4) Merge with navigation.
(5) Hierarchal sort (necessary for Final Print).
(6) Final Print (exoanded data set).
(7) Selection of all DART core recoveries.

PRECAUTIONS

(1) Navigation data sets on tape must be set up using 'prefio' before EDITCDI is called (example below).
(2) If a program run does not require navigation, 'prefio' may be skipped.
(3) In the example, interactive prompts are prefaced by a dash (-); system responses are prefaced by a bracket ([).
(4) The navigation data set is assumed to be on File 4 of ansi tape S6NCL2, produced by MAGNAVOX SATNAV system.

PROGRAM RUN

<u>TAPE SET-UP</u>

```
    prefio
     -Enter switch number, eg, 09
    10
     -Enter name for dummy segment (max. 6 chars):
    dum
     -Input or output action (i/o)?
    i
     [r 1605 0.552 7.780 126
    io attach file10 tape_ansi_ S6NCL2 -nm F4
     [Mounting volume S6NCL2 with no write ring.
```

PROGRAM EDITCDI

```
         [S6NCL2 mounted on tape_03
         [r 1607 1.180 12.224 174
        io open file10 sqi
         [r 1607 0.347 2.178 56

    INITIATE_EDITCDI
        bdafe$cdimain
         -EDIT CRUISE DATA INDEX
         -PROMPT FOR COMMANDS IS >>>>

    INPUT_DATASET
         -ENTER AT LEAST FIRST 4 CHARACTERS OF THE COMMAND ON THE
         NEXT LINE.
         ->>>>
        inpu
         -ENTER INPUT SEGMENT NAME - UP TO 32 CHARACTERS.
        s377eg
         -??IS DATASET IN FINAL PUNCH CARD FORMAT?? (t or f)
        f
         -??HAS NAV. BEEN MERGED WITH DATA SET?? (t or f)
        f
         -??IS TIME IN DECIMAL MINUTES?? (t or f) (l1)
        f
         -??HAVE HIERARCHAL NUMBERS/LADDER CODES BEEN ASSIGNED TO
         DATA LINE ENTRIES?? (t or f)(l1)
        f
         -662 RECORDS INPUT FOR THE CDI.

    AUTOMATIC_SEARCH
         ->>>>
        auto
         -??IS DATASET ALREADY TIMESORTED?? (t or f) (l1)
        f
         -??TIMESORT OUTPUT DESIRED?? (t or f)(l1)
        f
         -??HAVE HIERARCHAL NUMBERS/LADDER CODES BEEN ASSIGNED TO
         EVERY DATA LINE?? (t or f) (l1)
        f
         -??AUTOSORT OUTPUT DESIRED?? (t or f) (l1). IF FALSE, ONLY
         INDEX OF SORTED LINES PASSED.
        t
         -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
        s377auto

    TIME_SORT
         ->>>>
        sort
         -??SORT ON HIERARCHAL NUMBERS?? (t or f)(l1)
        f
         -??TIME/HIERSORT CARD IMAGE OUTPUT DESIRED?? (t or f)(l1) IF
         FALSE, ONLY INDEX OF SORTED LINES PASSED TO OTHER
         SUBROUTINES.
```

```
      t
       -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
      s377ts

   MERGE_NAVIGATION
       ->>>>
      merg
       -??CARD IMAGE DESIRED?? (t or f)(l1)
      t
       -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
      s377tsnm
       -??IS NAV. INPUT ON MAGNAVOX TAPE?? (t or f)
      t
       -***WARNING:  RECORD IS EARLIER THAN 1ST NAV. RECORD.  CDI
       DAY/TIME OF 180/1252.0, NAV DAY/TIME OF 180/1360.0

       -END OF CDI DATASET REACHED. RETURN TO COMMAND PROMPT.

   HIERARCHAL_SORT
       ->>>>
      sort
       -??SORT ON HIERARCHAL NUMBERS?? (t or f)(l1)
      t
       -??TIME/HIERSORT CARD IMAGE OUTPUT DESIRED?? (t or f)(l1) IF
       FALSE, ONLY INDEX OF SORTED LINES PASSSED TO OTHER
       SUBROUTINES.
      f

   FINAL_PRINT
       ->>>>
      prin
       -??IS DATA IN FINAL ORDER?? (t or f)(L1). IF NOT, CONTROL
       RETURNED TO COMMAND PROMPT -- CARD IMAGE OUTPUT RECOMMENDED.
      t
       -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
      Final_Prints3
       -??PAGE HEADERS DESIRED (SHIP,CHIEF SCI.,CRUISE ID.)?? (t or
       f)(l1)
      t
       -ENTER NUMBER OF CRUISE ID.'S IN DATASET. (i2) ****WARNING*
       IF CRUISE ID.'S >1, THEN DATA MUST BE FROM DIFF. TIME
       PERIODS OR HAVE BEEN SORTED SEPARATELY BEFORE MERGING.
      01
       -ENTER SHIP NAME(S), CHIEF SCIENTIST(S) AND CRUISE ID(S) IN
       2a8/2a8/a4 FORMAT.  I.E., GROUPS OF 3 VALUES, ONE VALUE PER
       LINE.
      R/V SEA SOUNDER
      __ALAN COOPER  (__ => TRAILING BLANKS TO CENTER NAME IN 16
       SPACES)
      __S3
       -??FINAL SEARCH OUTPUT?? (t or f)(l1) ***CAUTION* DATA MUST
       BE IN HIERARCHAL ORDER.  IF ""F"" ENTERED, FINAL TIMESORT
```

```
            OUTPUT ASSUMED.
           t
            -ENTER TODAY'S DATE:  I.E., 24 AUGUST 1944. (2a8)
           _4 OCTObER 1977
            -PRINT HAS BEEN OBTAINED  662 RECORDS OF OUTPUT

        SELECTIVE_SEARCH
            ->>>>
           sear
            -??CARD IMAGE OUTPUT DESIRED?? (t or f).  IF NOT, OUTPUT
            WILL BE ROUTED THRU FINPRN & WILL BE AFFIXED W/COLUMN
            HEADERS.
           t
            -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
           s377ser
            -ENTRIES REQUIRED BY SEARCH WILL BE PROMPTED BY >>.
            -ENTER NUMBER OF SEARCH SEQUENCES. (i2 W/LEADING ZEROS)
            ->>
           01
            -??INCORRECT ENTRY?? (t or f). IF TRUE, RE-ENTER.
           f
            -ENTER NUMBER OF SEARCH LEVELS PER SEARCH SEQUENCE. (i1)
            ->>
           2
            -??INCORRECT ENTRY?? (t or f).  IF TRUE, RE-ENTER.
           f
            -FOR EACH SEARCH LEVEL, ENTER THE INDEX OF VARIABLE S(INDEX)
            AND THE VALUE TO BE SEARCHED. (i2 RT. JUST. LEADING ZEROS;
            a4 LFT. JUST. TRAILING BLANKS)  NUMBER OF ENTRY PAIRS MUST =
            NUMBER OF SEARCH LEVELS.
            -??IS SEARCH VALUE AN HIERARCH. NO. OR DEPTH?? (t or f)  IF
            TRUE, USE i6 W/LEADING ZEROS INSTEAD OF a4.
           f
            -ENTER VALUES
           ->>
            02DART
            -??INCORRECT ENTRY?? (t or f). IF TRUE, RE-ENTER.
           f
            -??IS SEARCH VALUE AN HIERARCH. NO. OR DEPTH?? (t or f)  IF
            TRUE, USE i6 W/LEADING ZEROS INSTEAD OF a4.
            ->>
           f
            -ENTER VALUES
            ->>
           08Y
            -??INCORRECT ENTRY?? (t or f).IF TRUE, RE-ENTER.
           f
            -50 RECORDS FOUND FOR SEARCH SEQUENCE  1

        END_OF_PROGRAM_RUN
            ->>>>
           stop
```

```
[STOP
[fortran_io_: Close files?   yes
[r 1134 55.170 83.514 1077 level 2, 13
```

RESULTS

This program run will produce 5 data set segments in the user's working directory: s377auto, s377ts, s377tsnm, Final_Prints3, and s377ser.

V A R I A B L E   D I C T I O N A R Y

CDI RECORD VARIABLES

CARD_IMAGE_VARIABLES

s(1)	a3	Media code. Also in Data Statement array, MED.
s(2)	a4	Data code. Also in Data Statement array, COD.
s(3)	a1	Slave recorder, etc., marked with 'S'.
s(4)	a1	Start, marked with 'X'.
s(5)	a1	End, marked with 'X'.
s(6)	a1	On, marked with 'X'.
s(7)	a1	Off, marked with 'X'.
s(8)	a1	Recovery, marked with 'Y' or 'N'.
s(9)	i1	Depth - thousands
s(10)	i1	Depth - hundreds
s(11)	i1	Depth - tens
s(12)	i1	Depth - unit position
s(13)	a4	Cruise Locator - leading blanks if only 2 char.
s(14)	i2	Year
s(15)	a2	Area
dy	i3	Julian day
hr	i2	Hour of day (GMT)
mn	i2	Minutes of hour
rmn	f4.1	Decimal minutes
sc	i2	Seconds of minutes
lin	a4	Trackline number
sp	a4	Shot point or station number
isn	a4	Sequence number including tracklines, stations, samples, rolls, reels, etc. Also used for institution code.
idd(6)	6a4	Comments or data description
iu	a1	Depth units (i.e., 'm' for meters)

ADDITIONAL_VARIABLES

s(16)	a1	1st level Ladder Code - also in Data Statement l1
s(17)	a1	2nd level Ladder Code - also in Data Statement l2
s(18)	a1	3rd level Ladder Code - also in Data Statement l3
s(19)	a1	4th level Ladder Code - also in Data Statement l4
ic	i3	Position of Data Code in COD.
im	i2	Position of Media Code in MED.
ilt	i8	Latitude - decimal degrees expanded by 1000000.
iln	i9	Longitude - decimal degrees expanded by 1000000.

VARIABLES RELATED TO MEDIA CODES

		HM		
	JM	MG		
MED___________	IM	MH	II	JN
NAV(STAT/TRAK)	1	10	1	13
PER	2	12	3	-
EQP	3	13	4	-
SAM	4	8	-1	1
PAP	5	1	-8	2
MAG	6	2	-7	3
SYS	7	0	-9	4
OBS	8	0	-9	5
AVM	9	3	-6	6
PHO	10	9	0	7
PRN	11	4	-5	8
PUN	12	5	-4	9
MAP	13	0	-9	10
CRD	14	6	-3	11
DIG	15	7	-2	12
NAV(CRUZ/PORT)	16	11	2	13
				(etc)

==

COMPREHENSIVE DESCRIPTION AND PURPOSE

==

OVERALL

EDITCDI was written to organize, sort, edit and display data log entries which have been selectively keypunched from the data logs of cruises sponsored by the Office of Marine Geology, U.S.G.S., Menlo Park, CA. These data lines constitute a 'Cruise Data Index' (CDI) for a particular cruise, and they relate where data is collected to what type of data was collectd there and what recording medium, if any, was used to record the data.

Each CDI record contains a time, a 3-letter 'Media Code', a 4-letter 'Data Code', and, if the data is continuous underway data, there is also an indicator signaling whether the data has just started to be recorded/collected or has just stopped. The time is used to relate the data record to a navigational position for that time or it is used to interpolate a position from navigation at surrounding times. The 3-letter 'Media Code' is a mneumonic code indicating either the type of recording medium the data is being recorded on or, if it is not analog data, what the general category of the data is. The 4-letter 'Data Code' is also a mneumonic code but specifically identifies the type of data being collected.

By operating on these basic elements of a CDI data record, EDITCDI can produce both chronological lists of the data and lists grouped according to data type which are easily edited and amenable to clear display. It can also merge navigation positions with the data and make selections of single data types or combinations of data types to be used for making map-plots.

Detailed explanations and descriptions of shipboard data logging and annotating procedures as well as complete lists of current 'Data' and 'Media' Codes can be found in the Marine Data Information System documentation booklet.

CDIMAIN

CDIMAIN is the entry program for EDITCDI. It is used to interactively receive the commands which determine the type of manipulation to be performed on the data.

Possible commands are inpu, auto, sort, sear, merg, prin, punc, outp and stop and are prompted by >>>> on the user's terminal. Their use is described in the OPERATION section and in the subroutine descriptions below.

CDI MAIN also interactively prompts the user for information that sets up the necessary conditions for each successive manipulation of the data in a given program run. Required information includes questions about the current order and format of the data and what kind of output is desired.

SUBROUTINES

PROGRAM EDITCDI

There are 9 subroutines, 7 of which directly act on the data to input, organize, condense, or add to it in some way for output. Of the other two subroutines, one stores pertinent variables and data in Common Block or Data Statements used by the rest of the program, while the other produces a title for the 'Final Print' output-type. A functional description of each subroutine follows:

INCDI - Input of CDI Data Sets - Activated by inpu

FUNCTION

This subroutine contains the formats for the different types of CDI input data sets. Entry of the command inpu causes the program to branch to a section of CDI MAIN which asks the user about the type and condition of the input data set and then calls INCDI to read the data set into the computer. Five types of input are possible:

INPUT DATA CATEGORIES

1. Initial input from original CDI 80 column records with time in hours, minutes and seconds. After input, seconds are immediately converted to decimal minutes.

2. Input of 80 column records which have had time converted to hours and decimal minutes in a previous run of EDITCDI on the data (see 1, above).

3. Input of records which have had 'additional variables' added to the original 80 column records by the subroutine AUTOSR. The extra variables include a 4-letter 'Ladder Code', a 6 digit 'Hierarchal Number', and 2 numbers which locate the position of the Media and Data Codes in the program arrays MED and COD which correspond to the Media and Data Codes of the record. These 'additional variables' are explained in more detail in the notes on the subroutine AUTOSR.

4. Input from records which have had the 'additional variables' added to them by AUTOSR plus latitude and longitude added to them by NAVMER.

5. Input from condensed 80 column records created by the subroutine FINPUN. The record is a condensation of a CDI record which has had latitude and longitude and the other 'additional variables' added to it, however, instead of containing the actual Media, Data or Ladder codes, it has their array element position numbers converted to hexidecimal numbers. When such a record is read, these numbers are used to locate the Media, Data, and Ladder

codes and the Hierarchal number in their respective Data Statements and add them to the record.

When reading a condensed data set into the EDITCDI, the user will be asked whether he wants to see a listing of the expanded data set. If an answer of true or t is given, a listing of the data set will be written in his working directory under file09, which can then be printed out.

RESTRAINTS

If the input data set is greater than 2400 lines, INCDI will truncate excess lines. 2400 is the effective upper limit for the CDI data variable array sizes on the Honeywell system. Greater array size prevents compilation.

AUTOSR -Automatic Sort - Activated by auto

FUNCTIONS

AUTOSR performs two functions. First, it is designed to group CDI records by 'Media Code' types which are further subdivided into sections of the same 'Data Code' type. Visual editing and correction is easily performed on a CDI data set organized in this fashion. Second, each data line is assigned a 4-letter Ladder Code, a 'Hierarchal Number', and numbers which locate the position of the Media and Data codes and corresponding Hierarchal numbers in the Data Statement arrays containing their literal values located in BLOCK_DATA.

LADDER CODES

A Ladder Code consists of four letters. The first letter can be one of 4 possible letters and indicates whether the data record is geophysical (X), geological (G), hydrographic (H), or vessel (V) data. The second letter can be one of 16 possibilities (not listed here) which indicate either the type of recording medium the data is being recorded on or the general data category of the data record, eg., whether a physical sample (S) or a numerical data observation (O). The third letter identifies broad groupings of related data types such as all high resolution acoustic data (H) or all geologic sediment cores (C). The fourth letter identifies the specific data type of a CDI record. Each data type does not necessarily have a unique 3rd or 4th letter but these letters are always in unique combination with the first two letters. The Ladder Codes are useful because they enable the user to select out any combination or unique category of data lines from a CDI data set by using the command sear which activates the subroutine SELSCH.

HIERARCHAL NUMBERS

A Hierarchal Number is a 6 digit number assigned to each CDI data line according to its data type. When a CDI data set is sorted sequentially by this number, the data set is automatically arranged in a logically interrelated and fixed order. This order enhances readability and uniformity when used on data sets written out by the subroutine FINPRN or by the PRINT subroutine of the STATCDI program. This number can also be used in the subroutine SELSCH to select out various data categories.

MEDIA AND DATA CODE ARRAY POSITION NUMBERS

These numbers locate the position of a record's Media and Data codes in the Data Statement arrays MED and COD of the subroutine BLOCK_DATA. The numbers are then used internally in place of the Media and Data codes and can be arithmetically manipulated as well as used to locate other literal data relating to the media or data type such as Hierarchal Numbers and more complete descriptions of the data type.

INPUT

1. AUTOSR will not work on data lines containing latitude/longitude values unless input data set is the Final Punch condensed version.

2. Output is useless if input is not in timesort order. After auto command is issued, user will be asked if data is timesorted. An answer of false or f will cause the data set to pass through the subroutine THSORT before proceeding to AUTOSR. An option exists to output the timesort to the working directory before it goes through AUTOSR.

3. Media and Data codes used in CDI log entries which are not on official lists of such codes, will be flagged at the users terminal. Data lines containing these codes will not receive the 'additional variables'.

4. A data set which has been through AUTOSR but is missing 'additional variables' from some of its data lines can be run through AUTOSR again to pick up the missing variables - provided the conditions which prevented the assignment of the variables to the lines has been corrected.

OUTPUT

1. The order in the data created by AUTOSR is preserved in the index variable, NI. That is, each consecutive element of NI contains the line numbers of the record that is supposed to be in that particular consecutive position. The line number is the original position of the record in the data set as it was read in at the start of the program run. Any subsequent manipulations of the data in the same program run which are initiated by command from CDIMAIN will use the positions of the data lines passed to it by this index variable.

2. An option exists to write the output of AUTOSR to the users working directory. This does not preclude initiating further manipulation of the data set from CDIMAIN. The output will contain the 'additional variables' generated by AUTOSR and if the original input data set was a condensed version of a CDI which had navigation, navigation will be added to this output.

THSORT - Time and Hierarchal Sort - Activated sort/auto

FUNCTION

This subroutine was designed to organize an input data set in either chronological order or a sequential order based on 'Hierarchal Numbers' which are assigned by the subroutine AUTOSR. THSORT is activated by the command sort. It is also activated by auto if the data set to be input to AUTOSR has not been previously time-sorted.

TIME-SORT

After entering the command sort, the user will be asked if he wants to hierarchally sort the input data set. An answer of false or f will cause each input record's year, day, hour and minute values to be converted to a single decimal day value (1960 is taken as year zero). This number is then read into the sorting variable, SV, and is arranged in ascending sequential order. The original line numbers are placed in the same order in the index variable NI.

HIERARCHAL-SORT

An answer of true or t to this query will cause the 'Hierarchal Number' of each data line to be placed in the sorting variable, SV, and the data lines will be arranged in the same order as the sorted Hierarchal Numbers. Hierarchally sorting a CDI data set arranges the data lines in groups of data categories and also places the sequence of the groups in a predetermined order.

INPUT

All possible forms of a CDI data set can be fed into the time-sort part of THSORT, either an 80 column record or one that has had navigation and 'additional variables' added. Input to be hierarchally sorted must have had hierarchal numbers assigned to the data lines by AUTOSR. The input order of the data lines fed into the hierarchal sort part of this subroutine can be the order of the initial input to the program or can come from the order of an index variable, NI generated by AUTOSR or by a previous run of the data through the timesort part of THSORT.

OUTPUT

The option can be chosen to write the output of this subroutine to the user's working directory. Output can include 80 column records with no 'additional variables' and records with 'additional variables', or 'additional variables' and navigation. If output is not desired, the order of the data created by THSORT is preserved and passed on in the index variable, NI, for subsequent manipulations by other subroutines.

SELSCH - Selective Search - Activated by sear

FUNCTION

This subroutine is used to make selections of records from CDI data sets. Such a selection can be useful for creating master data sets containing a single data type from several cruises, or for separating particular types or categories of data-records from the CDI of a single cruise. Both types of data sets can then be used to make map-plots of the data locations, either for a single cruise or for several. An option also exists to print the selected records out in the highly readable format generated by the subroutine FINPRN.

SELECTION POSSIBILITIES

CDI data record selections can be made by searching for any one of the twenty record variables which are stored in the 20 elements of program variable 'S' when the CDI is read into the program. These 20 variables were judged to be the most useful ones for making meaningful selections from the CDI. They are listed in the Variable Dictionary section of this documentation. Selsch allows the user to select out more than one type of record at a time. Each process that has to be performed to select out all records of one type is called a search sequence. Each search sequence has one or more search operations necessary to cull out all of the records of a given category from the CDI data set. These individual search operations are called search levels.

EXAMPLES

For instance, a search sequence performed to select out all DART core sample attempts which had recoveries would require two separate search levels: one to select out the DART core sample attempts and another to select out the successful recoveries from the attempts. Another example is that the user might want to select out all records with a water depth of 3250 - 3259 meters. This would require a search sequence with three search levels. The first level would look for a 3 in the 'S' variable containing the thousands place S(9); the second level would look for a 2 in the 'S' variable containing the hundreds place, S(10); and the last search would be for a 5 in the 'S' variable that contains the tens place S(11).

STRATEGY

A logical strategy to employ when using SELSCH is to be sure that for each search sequence, the first search level selects out the variable which narrows the search spectrum most rapidly. As an example, it would be inefficient to use 'recoveries' as the first search level of a search sequence looking for DART core recoveries in a data set containing many hundreds of other sample recoveries. It would be best to search for DART cores at the first level. In the water depth example, the selection of 3 in the thousands place for the first search level narrows the search much more rapidly than if 5 in the tens place were used for the first search level.

USE

On entering the command SEAR, the user will be asked several questions including whether or not card-image output is desired, ie., if true, record output will be equivalent to record input; the number of search sequences to be performed; the number of search levels per search sequence; the index of 'S' for each search level, ie., its array element number for the particular variable to be searched; and the actual value of each 'S' to be searched. Replies to these questions in SELSCH will be prompted by >>.

It is important that 'S' indices and values are entered in pairs for each search level and that they are entered in the same order as the search levels of a particular search sequence. Entry of each block of 'S' indices and values should also follow the order of the search sequences. The indices and values of 'S' as well as their order of entry should be determined before the program run.

An additional feature of this subroutine is that the user will be asked if each value entered was correctly entered. An answer of false or f will cause the program to re-prompt for that particular entry. Finally, if a search is being performed on hierarchal numbers or depth, a slightly different entry format is used. Instead of A4 format, it will be I6, right justified with leading zeroes.

INPUT/OUTPUT

This subroutine will operate on each type of possible input to EDITCDI. It will also output each of these formats. Also, FINPRN can be called from SELSCH if a more readable output is desired.

<u>NAVMER</u> - Navigation Merge - Activated by merg

FUNCTION

This subroutine merges navigational positions from a separate navigation data set for a cruise with the scientific data records from the CDI of the cruise. Non-scientific CDI records will not take on navigational positions.

The merging is accomplished in two ways. <u>First</u>, a CDI record which has the same time as a particular navigation record will take on the navigational position of that navigation record. <u>Second</u>, if the time of a CDI record falls between the times of successive navigation records, a navigational position is linearly interpolated for the CDI record based on the time differentials between the navigation records and between the CDI record and the second navigation record.

INPUT

A Cruise Data Index must have had the 'additional variables' added to it by AUTOSR before it can be run through NAVMER. It must also be in chronological order. The user's job process will be returned to the command-prompt statement in CDIMAIN if these conditions are not met.

Input navigation data sets may reside on tapes generated from either the Marconi or Magnavox navigation systems or they may be in card format. If the navigation is on cards, the user should be prepared to indicate which of three possible formats the positions are in, whether they were entered with implied decimal points or not, or whether space was allowed for a negative latitude value.

OUTPUT

A merged CDI data set can be written to the user's working directory or another command can be given to perform further manipulations on it before it is finally written or displayed with corresponding navigation positions.

A navigational data set may have considerable gaps in it or it may end before the last of the CDI records. Messages to the terminal will indicate when CDI records are earlier than the first nav record and navigation values will be set to zero for these records. Messages will also indicate when the difference in time between a CDI record and each of the surrounding navigation records is greater than 2 hours. When this happens, interpolation will be skipped and the CDI record will be given a value of zero.

FINPRN - Final Print - Activated by prin

FUNCTION

This subroutine prints out a Cruise Data Index in a more organized and readable format that the initial index as keypunched from cruise logs. It translates and expands the Media and Data Codes and the start/end/on/off indicators , etc., of a CDI record to their English equivalents and displays them under clearly labeled columns.

POSSIBLE DISPLAYS

A Final-Print listing can be displayed in 4 possible ways. First, the data can be grouped according to type where each group is preceded by a descriptor signifying the data type of the group. The proper grouping is achieved by running an edited CDI data set through the hierarchal sort part of the THSORT subroutine. Second, the data may be displayed with the records in chronological order. In this case, the data type descriptors are absent as are the gaps in the listing separating groups. The data must be edited and run through the time-sort portion of THSORT before activating FINPRN. *Third*, and *fourth*, in each of the foregoing cases, a page header consisting of ship name, chief scientist, and Cruise Locator may be included or excluded from the top of each page. If it is excluded, the Cruise Locator will be added to each line of the listing.

In all four cases a title will be added to the listing. The title includes the Cruise Locator and Area Code, the ship name, the chief scientist, whether the listing is sorted by time or data type, and the date when the listing was generated. If the first record of an input CDI data set does not have the same cruise ID as indicated by the user in CDIMAIN, the user will have the option of aborting the run or continuing it, but the listing will lack a title and page headers.

USE CONSIDERATIONS

At the command prin, the user will be asked whether the input data set is in final order. Final order means it must have had the 'additional variables' added to it in AUTOSR, it must be a corrected CDI and it must be in either hierarchal or chronological order. If an answer of f is given, the process will return to the command prompt statment in CDIMAIN.

The user will also be asked whether page headers are desired and how many different cruise ID's there are in his data set. An input data set may consist of records with different cruise ID's. However, the records must be in groups of the same ID. This is achieved if records of differing ID's are also from different time periods or if records of the same ID are time or hierarchally sorted *before* being combined with record groups with different ID's. In either case, as FINPRN processes such a combined data set, it will print out a new title and start a new page for each different group it encounters.

For each ID group, the user will be asked to enter a ship name, name of chief scientist and a cruise ID before FINPRN is called up.

FINPRN will process input data sets with or without navigation points. It will also accept input directly from the SELSCH subroutine provided all the above conditions are met.

FINPUN - Final Punch - Activated by punc

PROGRAM EDITCDI

FUNCTION

This subroutine translates an edited CDI data set into one of two 80 column card image formats which can contain all the pertinent information from the original CDI records including latitude and longitude.

INHOUSE FORMAT

The first format is for inhouse use and serves as the storage format for edited Cruise Data Index files. A CDI record is condensed into 80 columns by using the position numbers of the Media and Data codes in MED and COD in place of the Media and Data Codes themselves. These numbers are further condensed by converting them to hexidecimal numbers. Also, the start/end/on/off indicator of a record is converted to a 1/2/3 or 4 and the decimal minute value is converted to an integer.

USE OF THE INHOUSE FORMAT

When a final punch inhouse data set is read back in to EDITCDI the hexidecimal numbers are translated back to the array position numbers of the Media and Data Codes in MED and COD. These numbers are then used to replace the actual Media, Data and Ladder Codes and the Hierarchal Number to the CDI record input variable set and the data set becomes the same as the one used to form the Final Punch inhouse data set.

NGSDC FORMAT

This format was designed for NGSDC but currently is not required by them. It retains the Ladder Codes and the actual Data Codes but drops the Media Codes. It also translates a record's start/end/on/off indicator to 1/2/3 or 4. Indication of start/end/on/off as well as sample recovery or no recovery and media or general data type is written out in the field of the record normally preserved for comments - except for PORT, CRUZ, EQP and PER records.

TITLE

TITLE is called from FINPRN every time a group of CDI records with a different cruise ID are encountered within the CDI data set being processed. It prints out a title on the page preceding the beginning of each new group. Information included in the title is the Cruise Locator and Area code; the cruise ship; the Chief Scientist; and a comment indicating whether the subsequent list has been organized by data type or chronologically. Also, included is the date the listing is generated.

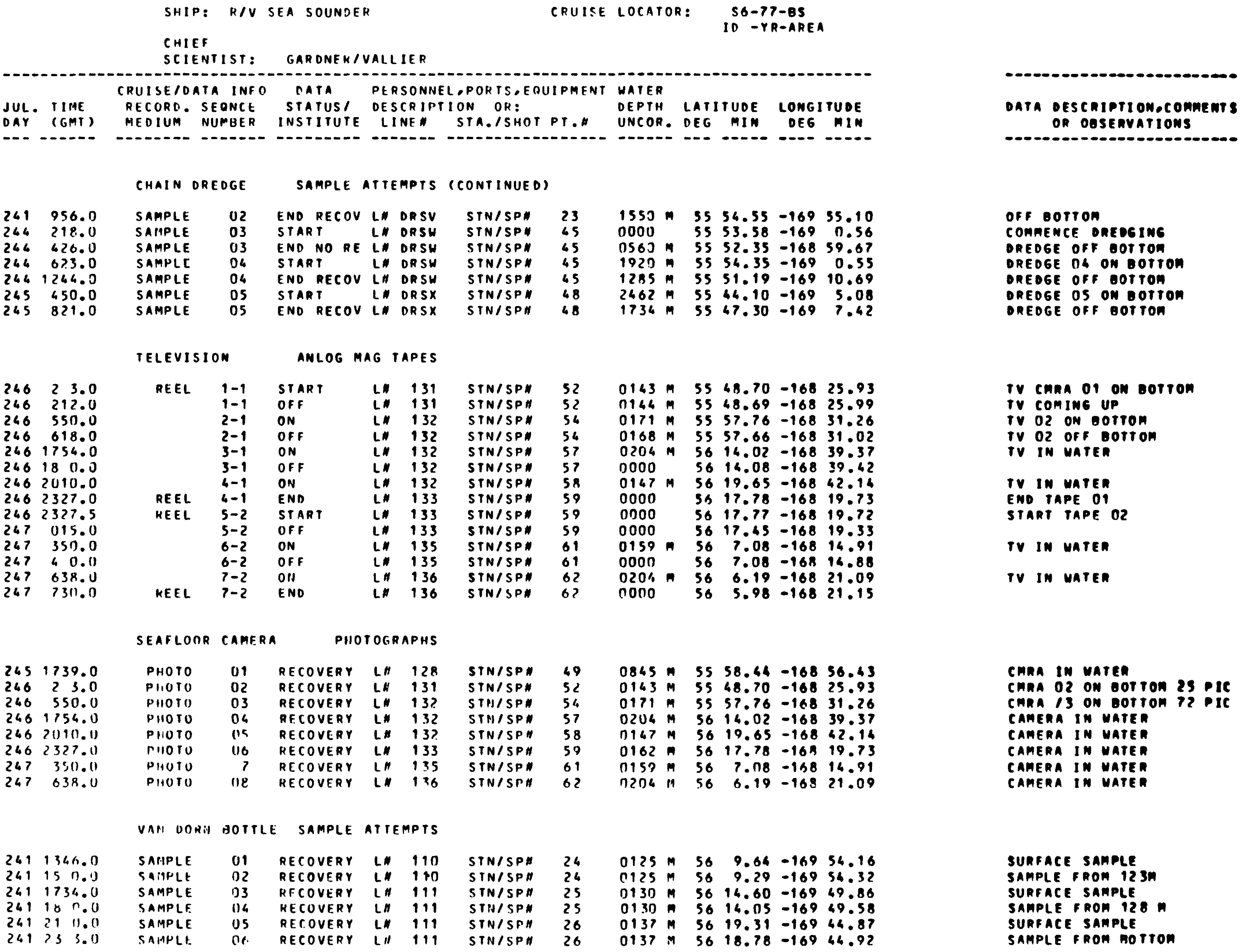
PROGRAM EDITCDI

BLOCK_DATA

This is called from CDIMAIN at the initiation of a program run. It is a Block Data subroutine and contains the Labeled Common and Data Statements used throughout the program.

```
229 1120-.0   72  10 SAM GRAV   13         ON BOTTOM, CORE 13          Y0166M   S677BS GSCG 300108  91  4  5619912-16833345
237  938.0 TEST  14 SAM GRAV   14         GRAVITY CORE 14 ON BOTT     Y1309M   S677BS GSCG 300108  91  4  5421100-16663906
238 1155.0  101  15 SAM GRAV   15         ON BOTTOM, GRAV CORE 15     Y0825M   S677BS GSCG 300108  91  4  5434194-16751842
238 1225.0  101  15 SAM GRAV   16         G-16 ON BOTTOM              Y0825M   S677BS GSCG 300108  91  4  5433600-16751563
238 1611.0  102  16 SAM GRAV   17         G-17 ON BOTTOM              N1195M   S677BS GSCG 300108  91  4  5437921-16808920
238 1641.0  102  16 SAM GRAV   18         G-18 ON BOTTOM              Y1195M   S677BS GSCG 300108  91  4  5437756-16808740
238 1717.0  102  16 SAM GRAV   19         G-19 ON BOTTOM              Y1190M   S677BS GSCG 300108  91  4  5437623-16808817
238 2219.0  103  17 SAM GRAV   20         GRAV CORE 20 ON BOTTOM      Y2230M   S677BS GSCG 300108  91  4  5476384-16870469
238 2319.0  103  17 SAM GRAV   21         GRAV CORE 21 ON BOTTOM      Y2224M   S677BS GSCG 300108  91  4  5476466-16870145
239  625.0  104  18 SAM GRAV   22         GRAV CORE 22 ON BOTTOM      Y2151M   S677BS GSCG 300108  91  4  5519565-16935063
239 1723.0  105  19 SAM GRAV   23         ON BOTTOM, G-23             Y2030M   S677BS GSCG 300108  91  4  5484007-17036963
240  025.0  106  20 SAM GRAV   24         GRAV CORE 24 ON BOTTOM      Y2900M   S677BS GSCG 300108  91  4  5552227-17002246
240  118.0  106  20 SAM GRAV   25         GRAV CORE 25 ON BOTTOM      Y2900M   S677BS GSCG 300108  91  4  5551610-17003206
240  621.0  107  21 SAM GRAV   26         GRAV CORE 26 ON BOTTOM      Y3158M   S677BS GSCG 300108  91  4  5571443-17080330
240  726.0  107  21 SAM GRAV   27         GRAV CORE 27 ON BOTTOM      Y3158M   S677BS GSCG 300108  91  4  5572246-17080805
240 11 0.0  108  22 SAM GRAV   28         G-28 ON BOTTOM              Y2630M   S677BS GSCG 300108  91  4  5593238-17047666
240 1158.0  108  22 SAM GRAV   29         G-29 ON BOTTOM              Y2648M   S677BS GSCG 300108  91  4  5593999-17053348
229 1212.0   72  10 SAM VANV   01         ON BOTTOM, VANV-1           N0166M   S677BS GSKV 301008 102  4  5620107-16833971
229 1239.0   72  10 SAM VANV   02         ON BOTTOM, VANV-2           N0166M   S677BS GSKV 301008 102  4  5619959-16833350
229 1254.0   72  10 SAM VANV   03         ON BOTTOM, VANV-3           N0166M   S677BS GSKV 301008 102  4  5619999-16833663
229 13 8.0   72  10 SAM VANV   04         ON BOTTOM, VANV-4           Y0166M   S677BS GSKV 301008 102  4  5620054-16833760
231 1419.0   84  11 SAM VANV  .05         ON BOTTOM, VANV-5           Y0062M   S677BS GSKV 301008 102  4  5503727-16449871
232  212.0   88  12 SAM VANV   06         VAN VEEN 06 ON BOTTOM       Y0076M   S677BS GSKV 301008 102  4  5480411-16498977
232 1412.0   92  13 SAM VANV   07         VAN VEEN 07 ON BOTTOM       Y0142M   S677BS GSKV 301008 102  4  5445603-16515885
226 2232.0   41  01 SAM CHAN   01    X    START DREDGE HAUL            0085M   S677BS GSDC 302008 113  4  5657751-16904459
226 2325.0   41  01 SAM CHAN   01     X   DREDGE OFF BOTTOM           Y0081M   S677BS GSDC 302008 113  4  5658174-16904789
241  751.0 DRSV  23 SAM CHAN   02    X    DREDGE ON BOTTOM             2150M   S677BS GSDC 302008 113  4  5591203-16995673
241  956.0 DRSV  23 SAM CHAN   02     X   OFF BOTTOM                  Y1550M   S677BS GSDC 302008 113  4  5590919-16991830
244  218.0 DRSW  45 SAM CHAN   03    X    COMMENCE DREDGING            0000    S677BS GSDC 302008 113  4  5589292-16900925
244  426.0 DRSW  45 SAM CHAN   03     X   DREDGE OFF BOTTOM           N0560M   S677BS GSDC 302008 113  4  5587248-16899446
244  623.0 DRSW  45 SAM CHAN   04    X    DREDGE 04 ON BOTTOM          1920M   S677BS GSDC 302008 113  4  5590576-16900914
244 1244.0 DRSW  45 SAM CHAN   04     X   DREDGE OFF BOTTOM           Y1285M   S677BS GSDC 302008 113  4  5585311-16917817
245  450.0 DRSX  48 SAM CHAN   05    X    DREDGE 05 ON BOTTOM          2462M   S677BS GSDC 302008 113  4  5573499-16908473
245  821.0 DRSX  48 SAM CHAN   05     X   DREDGE OFF BOTTOM           Y1734M   S677BS GSDC 302008 113  4  5578834-16912362
246  2 3.0  131  52 AVM TVIS  1-1    X    TV CMRA 01 ON BOTTOM         0143M   S677BS GTQI 305003 117  9  5581170-16843212
246  212.0  131  52 AVM TVIS  1-1        XTV COMING UP                 0144M   S677BS GTQI 305003 117  9  5581146-16843320
246  550.0  132  54 AVM TVIS  2-1       X TV 02 ON BOTTOM              0171M   S677BS GTQI 305003 117  9  5596275-16852103
246  618.0  132  54 AVM TVIS  2-1        XTV 02 OFF BOTTOM             0168M   S677BS GTQI 305003 117  9  5596103-16851708
246 1754.0  132  57 AVM TVIS  3-1       X TV IN WATER                  0204M   S677BS GTQI 305003 117  9  5623367-16865611
246 18 0.0  132  57 AVM TVIS  3-1        X                             0000    S677BS GTQI 305003 117  9  5623462-16865696
246 2010.0  132  58 AVM TVIS  4-1       X TV IN WATER                  0147M   S677BS GTQI 305003 117  9  5632755-16870228
246 2327.0  133  59 AVM TVIS  4-1     X   END TAPE 01                  0000    S677BS GTQI 305003 117  9  5629629-16832879
246 2327.5  133  59 AVM TVIS  5-2    X    START TAPE 02                0000    S677BS GTQI 305003 117  9  5629623-16832872
247  015.0  133  59 AVM TVIS  5-2        X.                            0000    S677BS GTQI 305003 117  9  5629087-16832221
247  350.0  135  61 AVM TVIS  6-2       X TV IN WATER                  0159M   S677BS GTQI 305003 117  9  5611808-16824854
247  4 0.0  135  61 AVM TVIS  6-2        X                             0000    S677BS GTQI 305003 117  9  5611797-16824795
247  638.0  136  62 AVM TVIS  7-2       X TV IN WATER                  0204M   S677BS GTQI 305003 117  9  5610311-16835149
247  730.0  136  62 AVM TVIS  7-2     X                                0000    S677BS GTQI 305003 117  9  5609972-16835249
245 1739.0  128  49 PHO CMRA   01         CMRA IN WATER               Y0845M   S677BS GPQJ 305109 118 10  5597403-16894046
246  2 3.0  131  52 PHO CMRA   02         CMRA 02 ON BOTTOM 25 PIC    Y0143M   S677BS GPQJ 305109 118 10  5581170-16843212
246  550.0  132  54 PHO CMRA   03         CMRA /3 ON BOTTOM 72 PIC    Y0171M   S677BS GPQJ 305109 118 10  5596275-16852103
246 1754.0  132  57 PHO CMRA   04         CAMERA IN WATER             Y0204M   S677BS GPQJ 305109 118 10  5623367-16865611
246 2010.0  132  58 PHO CMRA   05         CAMERA IN WATER             Y0147M   S677BS GPQJ 305109 118 10  5632755-16870228
246 2327.0  133  59 PHO CMRA   06         CAMERA IN WATER             Y0162M   S677BS GPQJ 305109 118 10  5629629-16832879
247  350.0  135  61 PHO CMRA    7         CAMERA IN WATER             Y0159M   S677BS GPQJ 305109 118 10  5611808-16824854
247  638.0  136  62 PHO CMRA   08         CAMERA IN WATER             Y0204M   S677BS GPQJ 305109 118 10  5610311-16835149
241 1346.0  110  24 SAM VAND   01         SURFACE SAMPLE              Y0125M   S677BS HSWV 400208 126  4  5616064-16990268
241 15 0.0  110  24 SAM VAND   02         SAMPLE FROM 123M            Y0125M   S677BS HSWV 400208 126  4  5615491-16990534
241 1734.0  111  25 SAM VAND   03         SURFACE SAMPLE              Y0130M   S677BS HSWV 400208 126  4  5624331-16983107
241 18 0.0  111  25 SAM VAND   04         SAMPLE FROM 128 M           Y0130M   S677BS HSWV 400208 126  4  5623410-16982641
```

FIGURE 25 LISTING OF CRUISE DATA INDEX AS IT IS KEYPUNCHED FROM LOGS BUT WITH LAT/LON, LADDER CODES, ETC., ADDED BY THE COMPUTER. THE DATA IS GROUPED ACCORDING TO TYPE



```
                  SHIP:  R/V SEA SOUNDER                          CRUISE LOCATOR:    S6-77-BS
                                                                                     ID -YR-AREA
                  CHIEF
                  SCIENTIST:    GARDNER/VALLIER
-------------------------------------------------------------------------------------------------------------------------------
              CRUISE/DATA INFO   DATA     PERSONNEL,PORTS,EQUIPMENT  WATER                                  
JUL.  TIME    RECORD. SEQNCE   STATUS/   DESCRIPTION  OR:            DEPTH  LATITUDE  LONGITUDE    DATA DESCRIPTION,COMMENTS
DAY   (GMT)   MEDIUM  NUMBER   INSTITUTE  LINE#    STA./SHOT PT.#    UNCOR. DEG  MIN   DEG  MIN         OR OBSERVATIONS
--- ------   -------- -------  --------- -------  ---------------   ------ ---  ----- ----  -----   -----------------------

               CHAIN DREDGE      SAMPLE ATTEMPTS (CONTINUED)

241  956.0     SAMPLE    02    END RECOV L# DRSV   STN/SP#    23     1550 M  55 54.55 -169 55.10    OFF BOTTOM
244  218.0     SAMPLE    03    START     L# DRSW   STN/SP#    45     0000    55 53.58 -169  0.56    COMMENCE DREDGING
244  426.0     SAMPLE    03    END NO RE L# DRSW   STN/SP#    45     0563 M  55 52.35 -168 59.67    DREDGE OFF BOTTOM
244  623.0     SAMPLE    04    START     L# DRSW   STN/SP#    45     1920 M  55 54.35 -169  0.55    DREDGE 04 ON BOTTOM
244 1244.0     SAMPLE    04    END RECOV L# DRSW   STN/SP#    45     1285 M  55 51.19 -169 10.69    DREDGE OFF BOTTOM
245  450.0     SAMPLE    05    START     L# DRSX   STN/SP#    48     2462 M  55 44.10 -169  5.08    DREDGE 05 ON BOTTOM
245  821.0     SAMPLE    05    END RECOV L# DRSX   STN/SP#    48     1734 M  55 47.30 -169  7.42    DREDGE OFF BOTTOM


               TELEVISION        ANLOG MAG TAPES

246  2 3.0      REEL    1-1    START     L#   131  STN/SP#    52     0143 M  55 48.70 -168 25.93    TV CMRA 01 ON BOTTOM
246  212.0              1-1    OFF       L#   131  STN/SP#    52     0144 M  55 48.69 -168 25.99    TV COMING UP
246  550.0              2-1    ON        L#   132  STN/SP#    54     0171 M  55 57.76 -168 31.26    TV 02 ON BOTTOM
246  618.0              2-1    OFF       L#   132  STN/SP#    54     0168 M  55 57.66 -168 31.02    TV 02 OFF BOTTOM
246 1754.0              3-1    ON        L#   132  STN/SP#    57     0204 M  56 14.02 -168 39.37    TV IN WATER
246 18 0.0              3-1    OFF       L#   132  STN/SP#    57     0000    56 14.08 -168 39.42
246 2010.0              4-1    ON        L#   132  STN/SP#    58     0147 M  56 19.65 -168 42.14    TV IN WATER
246 2327.0      REEL    4-1    END       L#   133  STN/SP#    59     0000    56 17.78 -168 19.73    END TAPE 01
246 2327.5      REEL    5-2    START     L#   133  STN/SP#    59     0000    56 17.77 -168 19.72    START TAPE 02
247  015.0              5-2    OFF       L#   133  STN/SP#    59     0000    56 17.45 -168 19.33
247  350.0              6-2    ON        L#   135  STN/SP#    61     0159 M  56  7.08 -168 14.91    TV IN WATER
247  4 0.0              6-2    OFF       L#   135  STN/SP#    61     0000    56  7.08 -168 14.88
247  638.0              7-2    ON        L#   136  STN/SP#    62     0204 M  56  6.19 -168 21.09    TV IN WATER
247  730.0      REEL    7-2    END       L#   136  STN/SP#    62     0000    56  5.98 -168 21.15


               SEAFLOOR CAMERA       PHOTOGRAPHS

245 1739.0      PHOTO    01    RECOVERY  L#   128  STN/SP#    49     0845 M  55 58.44 -168 56.43    CMRA IN WATER
246  2 3.0      PHOTO    02    RECOVERY  L#   131  STN/SP#    52     0143 M  55 48.70 -168 25.93    CMRA 02 ON BOTTOM 25 PIC
246  550.0      PHOTO    03    RECOVERY  L#   132  STN/SP#    54     0171 M  55 57.76 -168 31.26    CMRA /3 ON BOTTOM 72 PIC
246 1754.0      PHOTO    04    RECOVERY  L#   132  STN/SP#    57     0204 M  56 14.02 -168 39.37    CAMERA IN WATER
246 2010.0      PHOTO    05    RECOVERY  L#   132  STN/SP#    58     0147 M  56 19.65 -168 42.14    CAMERA IN WATER
246 2327.0      PHOTO    06    RECOVERY  L#   133  STN/SP#    59     0162 M  56 17.78 -168 19.73    CAMERA IN WATER
247  350.0      PHOTO     7    RECOVERY  L#   135  STN/SP#    61     0159 M  56  7.08 -168 14.91    CAMERA IN WATER
247  638.0      PHOTO    08    RECOVERY  L#   136  STN/SP#    62     0204 M  56  6.19 -168 21.09    CAMERA IN WATER


               VAN DORN BOTTLE   SAMPLE ATTEMPTS

241 1346.0     SAMPLE    01    RECOVERY  L#   110  STN/SP#    24     0125 M  56  9.64 -169 54.16    SURFACE SAMPLE
241 15 0.0     SAMPLE    02    RECOVERY  L#   110  STN/SP#    24     0125 M  56  9.29 -169 54.32    SAMPLE FROM 123M
241 1734.0     SAMPLE    03    RECOVERY  L#   111  STN/SP#    25     0130 M  56 14.60 -169 49.86    SURFACE SAMPLE
241 18 0.0     SAMPLE    04    RECOVERY  L#   111  STN/SP#    25     0130 M  56 14.05 -169 49.58    SAMPLE FROM 128 M
241 21 0.0     SAMPLE    05    RECOVERY  L#   111  STN/SP#    26     0137 M  56 19.31 -169 44.87    SURFACE SAMPLE
241 23 3.0     SAMPLE    06    RECOVERY  L#   111  STN/SP#    26     0137 M  56 18.78 -169 44.92    SAMPLE FROM BOTTOM
```

FIGURE 26 LISTING FROM A CRUISE DATA INDEX WHICH HAS BEEN TRANSLATED INTO A READABLE FORMAT BY THE SUBROUTINE FINPRN FROM THE PROGRAM EDITCDI

```
COMPILATION LISTING OF SEGMENT cdimain
Compileo by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 12/11/77  1136.1 pst Sun
    Options: table symbols

c********** EDITCDI MAINLINE **********
          logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm
          ,lout,liv,lrcp,lmgvx,lmarc
          real idt
          double precision sv,svk
          character*4 cmnd,iact,isn,rb
          character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,
          shp1,shp2
          character*32 segin,segout
          integer cod,dy,hi,hir,hm,hr,ic,im,iu,ios,l1,l2,l3,l4,n,ni,nsl,s,sp
          common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
          s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
          ,im(2400)
          common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
          nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
          ,lie,lpout,lnm,lout
          common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
          shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
          med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
          l4(158)
c
c*********COMMAND INPUT*********
          call block_data
          nocm=9
          write(6,2)
2         format(" EDIT CRUISE DATA INDEX")
10        write(6,12)
12        format(" PROMPT FOR COMMANDS IS >>>>"/" ENTER AT LEAST FIRST 4 C",%
          "HARACTERS OF THE COMMAND ON THE NEXT LINE.")
c ---- INPUT A COMMAND ----
c ---- ONE OF THE FOLLOWING COMMANDS IS ENTERED TO INITIATE PROGRAM ACTI
c         CDI DATA SET:  INPU, AUTO, SORT, SEAR, MERG, PRIN, PUNC, OUTP,
20        write(6,28)
28        format(" >>>>")
          read(5,36) iact
36        format(a4)
          do 44 i=1,nocm
          if(iact.eq.cmnd(i)) go to 60
44        continue
          write(6,52) iact,(cmnd(l),l=1,9)
52        format("0****ERROR: COMMAND ",a4," IS UNDEFINED. RECOGNIZABLE CO",%
          "MMANDS START WITH THE LETTERS"/" ",9(a4,2x)/"0 TRY AGAIN.")
          go to 20
c ---- BRANCH TO PROPER COMMAND ----
60        go to (100,202,304,406,508,610,712,814,999),i
c
c*********DATA INPUT*********
100       lsi=.false.
          lci=.false.
          lts=.false.
          lhs=.false.
          lnm=.false.
          write(6,115)
115       format(" ENTER INPUT SEGMENT NAME - UP TO 32 CHARACTERS.")
          read(5,120) segin
```

```
120       format(a32)
          call io ("attach","file08","vfile_",segin)
          call io ("open","file08","si")
          write(6,134)
134       format(" ??IS DATASET IN FINAL PUNCH CARD FORMAT?? (t or f)")
          read(5,158) lfp
          if(.not.lfp) go to 138
          lav=.true.
          ldt=.true.
          write(6,142)
          read(5,158) lnm
          write(6,136)
136       format(" ??IS PRINT LISTING OF INHOUSE DECK AFTER INPUT DESIRED??"
          /" (t or f) (l1)")
          read(5,158) lout
          go to 174
138       write(6,142)
142       format(" ??HAS NAV. BEEN MERGED WITH DATA SET?? (t or f)")
          read(5,158) lnm
          if(.not.lnm) go to 148
          lav=.true.
          ldt=.true.
          go to 162
148       write(6,150)
150       format(" ??IS TIME IN DECIMAL MINUTES?? (t or f) (l1)")
          read(5,158) ldt
158       format(l1)
162       write(6,166)
166       format(" ??HAVE HIERARCHAL NUMBERS/LADDER CODES BEEN ASSIGNED TO",%
          " DATA LINE ENTRIES?? (t or f)(l1)")
          read (5,158) lav
174       call incdi
          call io ("close","file08")
          call io ("detach","file08")
          go to 20
c
c*********AUTOMATIC TIMESORT/SEARCH*********
202       lhs=.false.
          write(6,232)
232       format(" ??IS DATASET ALREADY TIMESORTED?? (t or f) (l1)")
          read(5,158) lts
          if(lts) go to 282
          write(6,242)
242       format(" ??TIMESORT OUTPUT DESIRED?? (t or f)(l1). IF FALSE, ONL",%
          "Y INDEX OF SORTED LINES PASSED TO OTHER SUBROUTINES.")
          read(5,158) lci
          if(.not.lci) go to 262
          write(6,252)
252       format(" ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.")
          read(5,120) segout
          call io ("attach","file09","vfile_",segout)
          call io ("open","file09","so")
262       call thsort
          if(.not.lci) go to 266
          endfile 09
          call io ("close","file09")
          call io ("detach","file09")
266       lts=.true.
          lsi=.true.
          write(6,272)
```

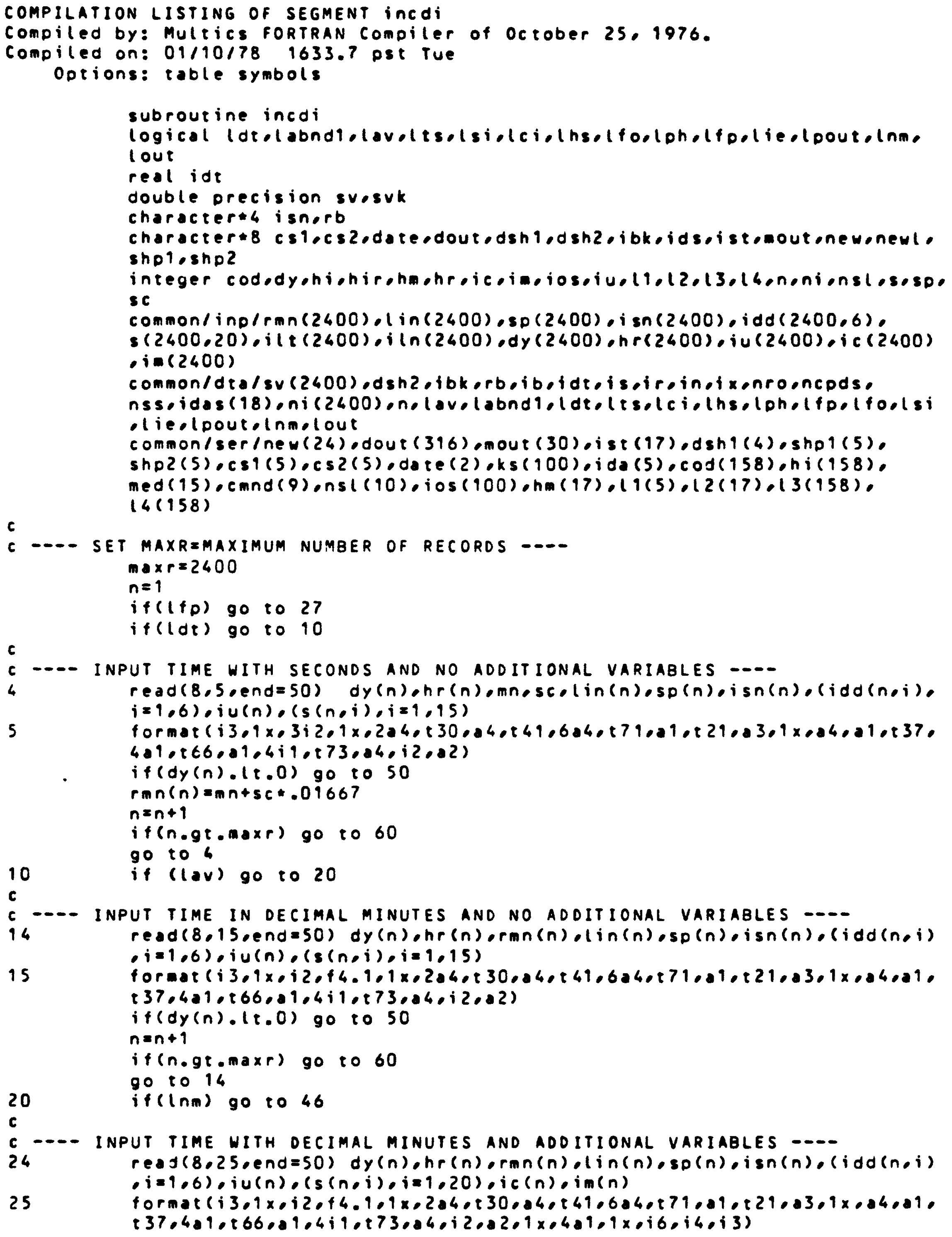
```
272       format(" ??HAVE HIERARCHAL NUMBERS/LADDER CODES BEEN ASSIGNED TO",%
          " EVERY DATA LINE?? (t or f)(l1)")
          read(5,158) lav
282       write(6,292)
292       format(" ??AUTOSORT OUTPUT DESIRED?? (t or f)(l1). IF FALSE, ONL",%
          "Y INDEX OF SORTED LINES PASSED.")
          read(5,158) lci
          if(.not.lci) go to 295
          write(6,252)
          read(5,120) segout
          call io ("attach","file09","vfile_",segout)
          call io ("open","file09","so")
295       call autosr
          if(.not.lci) go to 20
          endfile 09
          call io ("close","file09")
          call io ("detach","file09")
          go to 20
c
c*********SORT DATA BY TIME OR HIERARCHAL NUMBER*********
304       write(6,312)
312       format(" ??SORT ON HIERARCHAL NUMBERS?? (t or f)(l1)")
          read(5,158) lhs
          if(lhs) go to 334
c ---- LABND1 = T => INPUT DATA SET TRUNCATED AT 2401ST LINE ----
320       if(labnd1) write(6,328)
328       format(" --->NOTE: TIMESORTING WILL BE PERFORMED EVEN THOUGH ENT",%
          "IRE DATASET WAS NOT INPUT."/"        EXAMINE RESULTS CAREFULLY.")
          go to 368
c ---- ASSIGN HIERARCHAL NUMBERS S(N,20) TO SORTING VARIABLE SV(N) ----
334       if(lav) go to 340
          write(6,335)
335       format(" CONTROL RETURNED TO COMMAND PROMPT. "/" HIERARCHAL NUMB",%
          "ERS HAVE NOT BEEN ADDED TO DATASET.")
          go to 20
340       if(lsi) go to 352
          do 344 i=1,n
344       sv(i)=s(i,20)
          go to 368
352       do 360 i=1,n
          l=ni(i)
360       sv(i)=s(l,20)
c
368       write(6,376)
376       format(" ??TIME/HIERSORT CARD IMAGE OUTPUT DESIRED?? (t or f)(l1",%
          ") IF false, ONLY INDEX OF SORTED LINES PASSED TO OTHER"/"   SUB",%
          "ROUTINES.")
          read(5,158) lci
          if(.not.lci) go to 380
          write(6,252)
          read(5,120) segout
          call io ("attach","file09","vfile_",segout)
          call io ("open","file09","so")
380       call thsort
384       if(.not.lhs) lts=.true.
          lsi=.true.
          if(.not.lci) go to 20
          endfile 09
          call io ("close","file09")
          call io ("detach","file09")
          go to 20
c
c*********SELECTIVE SEARCH*********
406       write(6,409)
409       format(" ??CARD IMAGE OUTPUT DESIRED?? (t or f)."/" IF NOT, OUTP",%
          "UT WILL BE ROUTED THRU PRINT/FINPRNT & WILL BE AFFIXED W/COLUMN",%
          " HEADERS.")
          read(5,158) lci
          write(6,252)
          read(5,120) segout
          if(.not.lci) go to 412
          call io ("attach","file09","vfile_",segout)
          call io ("open","file09","so")
412       write(6,415)
415       format(" ENTRIES REQUIRED BY SEARCH WILL BE PROMPTED BY >>.")
          write(6,421)
421       format(" ENTER NUMBER OF SEARCH SEQUENCES. (I2 W/LEADING ZEROS)")
423       write(6,425)
425       format(" >>")
          read(5,430) nss
430       format(i2)
          write(6,432)
432       format(" ??INCORRECT ENTRY?? (t or f). IF TRUE, RE-ENTER.")
          write(6,425)
          read(5,158) lie
          if(lie) go to 423
          write(6,435)
435       format(" ENTER NUMBER OF SEARCH LEVELS PER SEARCH SEQUENCE. (I1)")
          ism=0
          do 441 i=1,nss
436       write(6,425)
          read(5,438) nsl(i)
438       format(i1)
          write(6,432)
          write(6,425)
          read(5,158) lie
          if(lie) go to 436
c ---- ISM IS THE TOTAL NUMBER OF SEARCH LEVELS ----
441       ism=nsl(i)+ism
442       write(6,445) ism
445       format(" FOR EACH SEARCH LEVEL, ENTER THE INDEX OF VARIABLE S(IN",%
          "DEX)"/" AND THE VALUE TO BE SEARCHED. (I2 RT. JUST. LEADING ZER",%
          "OS; a4 LFT. JUST. TRAILING BLANKS)"/" NUMBER OF ENTRY PAIRS MUS",%
          "T = ",i2,". (i2,1x,a4)")
          do 476 i=1,ism
448       write(6,451)
451       format(" ??IS SEARCH VALUE AN HIERARCH. # OR DEPTH?? (t or f)"/
          " IF TRUE, USE i6 W/LEADING ZEROS INSTEAD OF a4.")
          write(6,425)
          read(5,158) liv
          write(6,454)
454       format(" ENTER VALUES")
          if(liv) go to 460
          write(6,425)
          read(5,457) ios(i),ks(i)
457       format(i2,1x,a4)
          go to 466
460       write(6,425)
          read(5,463) ios(i),ks(i)
463       format(i2,i6)
```

```
466      if(lav) go to 472
         if(ios(i).lt.16) go to 472
         if(ios(i).gt.20) go to 472
         write(6,468)
468      format(" ***ERROR: LADDER CODES & HIERARCHAL #S NOT PRESENT IN D",%
         "ATA SET."/" ",13x,"IS RETURN TO COMMAND PROMPT DESIRED?? (t or ",%
         "f)")
         write(6,425)
         read(5,158) lrcp
         if(.not.lrcp) go to 472
         go to 10
472      write(6,432)
         write(6,425)
         read(5,158) lie
         if(lie) go to 448
476      continue
         call selsch (segout)
         go to 20
c
c*********NAVIGATION MERGE*********
508      if(lav) go to 510
         write(6,509)
509      format(" RETURN TO COMMAND PROMPT. USE -AUTO- TO ACQUIRE ARRAY I",%
         "NDEX VARIABLES.")
         go to 20
510      if(lts) go to 516
         write(6,232)
         read(5,158) lts
         if(lts) go to 516
         write(6,512)
512      format(" RETURN TO COMMAND PROMPT. CDI DATASET MUST BE IN TIMESO",%
         "RT ORDER FOR NAV. MERGE.")
                   go to 20
516      write(6,520)
520      format(" ??CARD IMAGE OUTPUT DESIRED?? (t or f) (l1)")
         read(5,158) lci
         if(.not.lci) go to 526
         write(6,252)
         read(5,120) segout
         call io ("attach","file09","vfile_",segout)
         call io ("open","file09","so")
526      write(6,530)
530      format(" ??IS NAV. INPUT ON MAGNAVOX TAPE?? (t or f)")
         read(5,158) lmgvx
         if(lmgvx) go to 550
         write(6,535)
535      format(" ??IS NAV. INPUT ON MARCONI TAPE?? (t or f)")
         read(5,158) lmarc
        if(lmarc) go to 550
         write(6,540)
540      format(" TYPE IN CODE NUMBER FOR NAVIGATION FORMAT (I1)"/"  1--S",%
         "TART IN COL. 54 WITH DECIMAL POINTS"/"  2--START IN COLUMN 54 W",%
         "ITH DECIMAL POINTS BUT NO PROVISION FOR NEGATIVE LATITUDES"/"  ",%
         "3--START IN COLUMN 55 WITH NO DECIMAL POINTS")
         read (5,438) kfor
550      call navmer (lmarc,lmgvx,kfor)
         lnm=.true.
         if(.not.lci) go to 20
         endfile 09
         call io ("close","file09")
```

```
          call io ("detach","file09")
          go to 20
c
c*********FINAL PRINT*********
610       if(lav) go to 614
          write(6,509)
          go to 20
614       write(6,615)
615       format(" ??IS DATA IN FINAL ORDER?? (t or f)(l1). IF NOT, CONTRO",%
          "L RETURNED TO COMMAND PROMPT -- CARD IMAGE OUTPUT RECOM-"/"   M",%
          "ENDED.")
          read(5,158) lfo
          if(.not.lfo) go to 20
          write(6,252)
          read(5,120) segout
          call io ("attach","file42","vfile_",segout)
          call io ("open","file42","so")
          write(6,626)
626       format(" ??PAGE HEADERS DESIRED (SHIP,CHIEF SCI.,CRUISE ID.)?? (",%
          "t OR f)(l1)")
          read(5,158) lph
          write(6,634)
634       format(" ENTER NUMBER OF CRUISE ID. S IN DATASET. (I2)"/" ****WA",%
          "RNING* IF CRUISE ID.S >1, THEN DATA MUST BE FROM DIFF. TIME PER",%
          "IODS OR HAVE BEEN SORTED SEPARATELY BEFORE"/"    MERGING.")
          read(5,430) ncpds
          write(6,642)
642       format(" ENTER SHIP NAME(S), CHIEF SCIENTIST(S) AND CRUISE ID(S)",%
          " IN 2a8/2a8/a4 FORMAT."/" I.E., GROUPS OF 3 VALUES, ONE VALUE E",%
          "NTERED PER LINE.")
          do 650 i=1,ncpds
650       read(5,652) shp1(i),shp2(i),cs1(i),cs2(i),ida(i)
652       format(2a8/2a8/a4)
660       if(lts.or.lhs) go to 664
          write(6,662)
662       format(" ??FINAL SEARCH OUTPUT?? (t or f)(l1)"/" ***CAUTION* DAT",%
          "A MUST BE IN HIERARCHAL ORDER.  IF ""f"" ENTERED, FINAL TIMESORT OUTPU%
          T ASSUMED.")
          read(5,158) lhs
664       write(6,666)
666       format(" ENTER TODAYS DATE; I.E., 24 AUGUST 1944. (2A8)")
          read(5,668) date(1),date(2)
668       format(2a8)
          call finprn
          endfile 42
          call io ("close","file42")
          call io ("detach","file42")
          if(nro.eq.0) go to 20
          write(6,674) nro
674       format("0PRINT HAS BEEN ATTAINED"/" ",i4,3x,"RECORDS OF OUTPUT")
          go to 20
c
c*********FINAL PUNCH:  NGSDC OR OER*********
712       if(lav) go to 715
          write(6,509)
          go to 20
715       write(6,716)
716       format(" ??FINAL CDI DECK FOR OUTSIDE O.M.G. DESIRED?? (t or f)."/
          " IF FALSE, FINAL INHOUSE DECK WILL BE GENERATED.")
          read(5,158) lpout
```

```
            if(lpout) go to 721
            if(lts) go to 719
            write(6,718)
718         format(" ??IS INPUT DATASET IN TIMESORT ORDER?? (t or f)")
            read(5,158) lts
            if(.not.lts) go to 727
719         if(.not.lav) go to 727
            go to 733
721         if(lhs) go to 733
            write(6,724)
724         format(" ??IS INPUT DATASET IN HIERARCHAL ORDER?? (t or f)")
            read(5,158) lhs
            if(lhs) go to 733
727         write(6,730)
730         format("0CONTROL RETURNED TO COMMAND PROMPT."/" INPUT MUST HAVE ",%
            "BEEN THRU AUTO, AND:"/" -INHOUSE DECK MUST BE IN TIMESORT ORDER."/" %
            -OUTSIDE DECK MUST BE IN HIERARCHAL ORDER.")
            go to 20
733         write(6,252)
            read(5,120) segout
            call io ("attach","file12","vfile_",segout)
            call io ("open","file12","so")
            call finpun
            endfile 12
            call io ("close","file12")
            call io ("detach","file12")
            go to 20
c
c*********OUTPUTS CONTENTS OF LINE INDEX VARIABLE*********
814         if(lsi) go to 822
            write(6,818)
818         format(" RETURN TO COMMAND PROMPT. ACCEPTS ONLY INDEX VARIABLE O",%
            "F DATA LINE POSITIONS.")
            go to 20
822         write(6,252)
            read(5,120) segout
            call io ("attach","file09","vfile_",segout)
            call io ("open","file09","so")
            do 848 i=1,n
            j=ni(i)
            if(lav) go to 834
            write(9,830)  dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1
            ,6),iu(j),(s(j,l),l=1,15)
830         format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
            t37,4a1,t66,a1,4i1,t73,a4,i2,a2)
            go to 848
834         if(lnm) go to 844
838         write(9,840)  dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1
            ,6),iu(j),(s(j,l),l=1,20),ic(j),im(j)
840         format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
            t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3)
            go to 848
844         write(9,846)  dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1
            ,6),iu(j),(s(j,l),l=1,20),ic(j),im(j),ilt(j),iln(j)
846         format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
            t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
848         continue
            endfile 09
            call io ("close","file09")
            call io ("detach","file09")
```

```
    850         go to 20
    999         stop
                end
```



```
COMPILATION LISTING OF SEGMENT incdi
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1633.7 pst Tue
    Options: table symbols

      subroutine incdi
      logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm,
      lout
      real idt
      double precision sv,svk
      character*4 isn,rb
      character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,
      shp1,shp2
      integer cod,dy,hi,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nsl,s,sp,
      sc
      common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
      s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
      ,im(2400)
      common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
      nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
      ,lie,lpout,lnm,lout
      common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
      shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
      med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
      l4(158)
c
c ---- SET MAXR=MAXIMUM NUMBER OF RECORDS ----
      maxr=2400
      n=1
      if(lfp) go to 27
      if(ldt) go to 10
c
c ---- INPUT TIME WITH SECONDS AND NO ADDITIONAL VARIABLES ----
4     read(8,5,end=50)  dy(n),hr(n),mn,sc,lin(n),sp(n),isn(n),(idd(n,i),
      i=1,6),iu(n),(s(n,i),i=1,15)
5     format(i3,1x,3i2,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,t37,
      4a1,t66,a1,4i1,t73,a4,i2,a2)
      if(dy(n).lt.0) go to 50
      rmn(n)=mn+sc*.01667
      n=n+1
      if(n.gt.maxr) go to 60
      go to 4
10    if (lav) go to 20
c
c ---- INPUT TIME IN DECIMAL MINUTES AND NO ADDITIONAL VARIABLES ----
14    read(8,15,end=50) dy(n),hr(n),rmn(n),lin(n),sp(n),isn(n),(idd(n,i)
      ,i=1,6),iu(n),(s(n,i),i=1,15)
15    format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
      t37,4a1,t66,a1,4i1,t73,a4,i2,a2)
      if(dy(n).lt.0) go to 50
      n=n+1
      if(n.gt.maxr) go to 60
      go to 14
20    if(lnm) go to 46
c
c ---- INPUT TIME WITH DECIMAL MINUTES AND ADDITIONAL VARIABLES ----
24    read(8,25,end=50) dy(n),hr(n),rmn(n),lin(n),sp(n),isn(n),(idd(n,i)
      ,i=1,6),iu(n),(s(n,i),i=1,20),ic(n),im(n)
25    format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
      t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3)
```

```
          if(dy(n).lt.0) go to 50
          n=n+1
          if(n.gt.maxr) go to 60
          go to 24
c
c ---- INPUT FROM FINAL PUNCH CARDS ----
27        read(8,28,end=50) dy(n),hr(n),rmn(n),lin(n),sp(n),imx,icx2,icx1,
          s(n,3),isn(n),ista,(idd(n,i),i=1,6),(s(n,i),i=8,12),iu(n),ilt(n),
          iln(n),(s(n,i),i=13,15)
28        format(i3,i2,f3.1,2a4,a1,i1,a1,a1,a4,i1,6a4,a1,4i1,a1,i8,i9,a4,i2
          ,a2)
          s(n,4)=ib
          s(n,5)=ib
          s(n,6)=ib
          s(n,7)=ib
c ---- CONVERT ISTA WHICH CONTAINS VALUES OF 0, 1, 2, 3 OR 4 (CORRESPONDING
         TO: NO ENTRY, START, END, ON, OFF) TO LITERAL 'X' VALUE IN REQUISITE
         POSITIONS OF CDI VARIABLE, 'S' ----
          if(ista.eq.0) go to 29
          l=ista+3
          s(n,l)=ix
29        jm=30
          jc=30
c ---- CONVERT HEXDEC. NUMBERS IMX, ICX1 TO BASE 10.  CONVERTED IMX = MEDIA
         CODE POSITION NUMBER JM, IN DATA STATEMENT ARRAY VARIABLE, MED ----
          do 30 i=1,15
          if(imx.eq.idas(i)) jm=i
          if(icx1.eq.idas(i)) jc=i
          if((jm+jc).lt.31) go to 34
30        continue
          if(icx1.eq.idas(18)) go to 33
          write(6,32)
32        format("0**INPUT ERROR** RETURN TO COMMAND PROMPT.")
          return
33        jc=0
c ---- DETERMINE DATA CODE POSITON NUMBER JC IN DATA STATEMENT ARRAY VAR -
         IABLE, COD, BY ADDING 16 X ICX2 TO CONVERTED ICX1 ----
34        jc=icx2*16+jc
c ---- REPLACE LITERAL MEDIA AND DATA CODE VARIABLES FROM MED AND COD IN
         CDI VARIABLE, 'S', USING JM AND JC ----
          s(n,1)=med(jm)
          s(n,2)=cod(jc)
          if(jc.lt.3) jm=16
c ---- USING JM AND JC, REPLACE LADDER CODES AND HIERARCHAL NUMBERS IN
         REQUISITE POSITIONS OF CDI VARIABLE, S ----
          s(n,17)=l2(jm)
          s(n,18)=l3(jc)
          s(n,19)=l4(jc)
          s(n,20)=hi(jc)+hm(jm)
          if(jc.lt.43) go to 36
          if(jc.lt.90) go to 38
          if(jc.lt.124) go to 40
          s(n,16)=l1(4)
          if(jm.eq.3) s(n,20)=104300
          go to 42
36        s(n,16)=l1(3)
          if(jm.eq.3) s(n,20)=104000
          go to 42
38        s(n,16)=l1(1)
          if(jm.eq.3) s(n,20)=104100
          go to 42
40        s(n,16)=l1(2)
          if(jm.eq.3) s(n,20)=104200
42        im(n)=jm
          ic(n)=jc
c ---- IF LOUT = T, BLOWN-BACK CONDENSED FINAL PUNCH DATA SET IS WRITTEN TO
c        WORKING DIRECTORY ----
          if(lout) write(9,44) dy(n),hr(n),rmn(n),lin(n),sp(n),isn(n),(idd(n
          ,i),i=1,6),iu(n),(s(n,i),i=1,20),ic(n),im(n),ilt(n),iln(n)
44        format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
          t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
          if(dy(n).lt.0) go to 50
          n=n+1
          if(n.gt.maxr) go to 60
          go to 27
c
c ---- INPUT FROM NAV. MERGED CDI DATASET ----
46        read(8,47,end=50) dy(n),hr(n),rmn(n),lin(n),sp(n),isn(n),(idd(n,i)
          ,i=1,6),iu(n),(s(n,i),i=1,20),ic(n),im(n),ilt(n),iln(n)
47        format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
          t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
          if(dy(n).lt.0) go to 50
          n=n+1
          if(n.gt.maxr) go to 60
          go to 46
50        n=n-1
          write(6,51) n
51        format(" ",i4," RECORDS INPUT FOR THE CDI.")
          labnd1=.false.
          return
60        n=n-1
c ---- IF INPUT DATASET > 2400 RECORDS, DATA SET TRUNCATED AT 2400TH RECORD
c        AND LABND1 SET = TRUE ----
          write(6,61) dy(n),hr(n),rmn(n),lin(n),sp(n),isn(n),(idd(n,i),i=1,6
          ),iu(n),(s(n,i),i=1,15)
61        format("0**** ERROR:  MORE THAN 2400 RECORDS ENCOUNTERED.  2400T",%
          "H RECORD WAS:  "/i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,
          a3,1x,a4,a1,t37,4a1,t66,a1,4i1,t73,a4,i2,a2/"0  DECREASE DATASET",%
          " OR REVISE DIMENSIONS.")
          labnd1=.true.
          return
          end
```

```
COMPILATION LISTING OF SEGMENT autosr
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1634.1 pst Tue
    Options: table symbols

      subroutine autosr
      logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm,
      lout,lmm
      real idt
      double precision sv,svk
      character*4 isn,rb
      character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,
      shp1,shp2
      integer cod,dy,hi,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nj,nk,
      nsl,s,sp
      common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
      s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
      ,im(2400)
      common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
      nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
      ,lie,lpout,lnm,lout
      common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
      shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
      med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
      l4(158)
      dimension nk(2400),nj(2400)
c
c ---- INITIALIZE CONDITIONS AND VARIABLES ----
      mk=0
      no=0
      jm=0
c ---- SET M = TO NUMBER OF DATA LINES IN INPUT DATA SET ----
      m=n
c ---- INITIALIZE INDEX VARIABLE NJ WHICH IS USED INTERNALLY TO THIS SUB-
c          ROUTINE ----
      do 10 i=1,m
      nj(i)=i
      if(lsi) nj(i)=ni(i)
10    continue
20    if(m.eq.0) go to 470
      lmm=.true.
      jj=0
      j=nj(1)
      if(.not.lav) go to 30
      jm=im(j)
      go to 60
c
c ---- TRIES TO MATCH MEDIA CODE OF INPUT RECORD TO THOSE STORED IN DATA
c          STATMENT ARRAY VARIABLE, MED ----
30    do 40 jm=1,15
      if(s(j,1).eq.med(jm)) go to 60
40    continue
      write(6,50) s(j,1)
50    format(" NO SUCH MEDIA CODE-->",a3,"  CORRECT S(1),L2,LAST DIGIT",%
      " IN S(17), AND IM.")
      jm=17
      mk=1
c ---- LMM = FALSE => RECORD MEDIA CODE DOES NOT MATCH PROGRAM MEDIA CODES IN
c          MED ----
      lmm=.false.
```

```
60        if(m.eq.1) go to 110
          nk(1)=j
          kk=m
          m=0
          jj=1
c
c ---- SELECTS OUT ALL RECORDS WITH THE SAME MEDIA CODE AS FOUND IN STATEMENT
c        30.  THEIR DATA LINE POSITION NUMBERS ARE STORED IN THE INDEX
c        VARIABLE NJ AS A WORKING DATA SET.  THE DATA LINE POSITION NUMBERS
c        OF REMAINING RECORDS ARE STORED IN INDEX VARIABLE NK TO AWAIT NEXT
c        MEDIA CODE SEARCH. ----
70        do 90 i=2,kk
          l=nj(i)
          if(s(j,1).ne.s(l,1)) go to 80
          jj=jj+1
          nk(jj)=l
          go to 90
80        m=m+1
          nj(m)=l
90        continue
c ---- JJ = 0 INDICATES ALL RECORDS OF A GIVEN MEDIA CODE SELECTION HAVE BEEN
c        ASSIGNED ADDITIONAL VARIABLES AND HAVE BEEN WRITTEN OUT.  A NEW MEDIA
c        CODE SELECTION WILL BE MADE ----
100       if(jj.eq.0) go to 20
          j=nk(1)
c ---- IF ADDITIONAL VARIABLES HAVE PREVIOUSLY BEEN ASSIGNED TO CDI DATASET,
c        ADDITIONAL VARIABLE SECTION SKIPPED ----
110       if(.not.lav) go to 120
          go to 310
120       kc=0
c
c
c ---- OPERATING INITIALLY WITH THE DATA CODE OF THE FIRST RECORD OF A NEW
c        MEDIA CODE SELECTION THEN WITH THE FIRST DATA CODE ENCOUNTERED DIF-
c        FERENT THAN THE ONE PREVIOUSLY SELECTED OUT OF THE MEDIA CODE SELEC-
c        TION AT STATEMENT 370, THIS SECTION DETERMINES THE POSITION OF
c        RECORD'S DATA CODE IN THE DATA STATEMENT VARIABLE COD, AND ASSIGNS
c        THE CORRESPONDING LADDER CODES AND HIERARCHAL NUMBERS ----
c
c ---- LOCATES POSITION OF DATA CODE IN COD AND ASSIGNS FIRST LADDER CODE TO
c        FIRST SINGLE RECORD OF EACH NEW DATA CODE ENCOUNTERED IN MEDIA CODE
c        SELECTION ----
          if(jm.lt.4) go to 190
          if(jm.eq.4) go to 170
c ---- GEOPHYSICAL DATA ----
130       s(j,16)=l1(1)
          kd=2
          do 140 i=43,89
          if(s(j,2).eq.cod(i)) go to 250
140       continue
c ---- HYDROGRAPHIC DATA ----
          s(j,16)=l1(4)
          kd=4
          do 150 i=124,158
          if(s(j,2).eq.cod(i)) go to 250
150       continue
160       if(kc.eq.0) go to 190
c ---- GEOLOGIC DATA ----
170       s(j,16)=l1(2)
          kd=3
          do 180 i=90,123
          if(s(j,2).eq.cod(i)) go to 250
180       continue
c ---- VESSEL DATA ----
190       s(j,16)=l1(3)
          kd=1
          do 200 i=1,42
          if(s(j,2).eq.cod(i)) go to 230
200       continue
          kc=kc+1
          if(.not.lmm) go to 210
          if(kc.lt.2) go to 130
210       write(6,220) s(j,2)
220       format(" NO SUCH DATA CODE-->",a4,".  CORRECT S(2),L1,L3,L4, AND",%
          " IC.")
          s(j,16)=ib
          s(j,20)=0
          i=i-1
          go to 300
230       if(2-i) 250,240,240
c ---- SETS JM = 16 IF DATA CODE IS CRUZ OR PORT ----
240       jm=16
c
c ---- ASSIGNS HIERARCHAL CODES TO FIRST SINGLE RECORD OF EACH NEW DATA
c        ENCOUNTERED ----
250       s(j,20)=hi(i)+hm(jm)
          if(jm.ne.3) go to 300
          go to (260,270,280,290),kd
260       s(j,20)=104000
          go to 300
270       s(j,20)=104100
          go to 300
280       s(j,20)=104200
          go to 300
290       s(j,20)=104300
c
c ---- ASSIGNS SECOND, THIRD AND FOURTH LADDER CODES TO FIRST SINGLE RECORD
c        OF EACH NEW DATA CODE ENCOUNTERED ----
300       s(j,19)=l4(i)
          s(j,18)=l3(i)
          s(j,17)=l2(jm)
          im(j)=jm
          if(.not.lmm) im(j)=0
          ic(j)=i
          if(i.eq.42) ic(j)=0
310       if(jj-1) 420,430,320
320       kk=jj-1
          jj=0
          no=no+1
          ni(no)=j
          if(.not.lci) go to 370
330       if(.not.lnm) go to 350
c
c ---- WRITES OUT FIRST LINE OF EACH NEW DATA CODE ----
          write(9,340) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
          6),iu(j),(s(j,l),l=1,20),ic(j),im(j),ilt(j),iln(j)
340       format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
          t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
          go to 370
350       write(9,360) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
             6),iu(j),(s(j,l),l=1,20),ic(j),im(j)
    360      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
             t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3)
    c
    c ---- STARTING WITH THE DATA CODE OF THE FIRST RECORD OF EACH NEW MEDIA CODE
    c         SELECTION, THIS SECTION SELECTS OUT FROM THE MEDIA CODE SELECTION
    c         ALL RECORDS WITH DATA CODES THAT CORRESPOND TO THE FIRST DATA CODE
    c         ENCOUNTERED AFTER THE PREVIOUS DATA CODE SELECTION ----
    370      do 410 i=1,kk
             l=nk(i+1)
             jj=jj+1
             nk(jj)=l
             if(s(j,2).ne.s(l,2)) go to 410
    c ---- JJ IS COUNT OF THE RECORDS REMAINING IN A MEDIA CODE SELECTION AFTER A
    c         PARTICULAR DATA CODE HAS BEEN SELECTED OUT ----
             jj=jj-1
             if(lav) go to 380
    c ---- ASSIGNS LADDER CODES, HIERARCHAL NUMBERS AND NUMBERS POSITIONING MEDIA
    c         AND DATA CODES IN MED AND COD TO ALL RECORDS WITH THE SAME DATA
    c         CODE PREVIOUSLY SEARCHED OUT IN THE SECTION STARTING WITH STATEMENT
    c         110 ----
             s(l,16)=s(j,16)
             s(l,17)=s(j,17)
             s(l,18)=s(j,18)
             s(l,19)=s(j,19)
             s(l,20)=s(j,20)
             im(l)=im(j)
             ic(l)=ic(j)
    380      no=no+1
             ni(no)=l
    c
    c ---- WRITES OUT REMAINING RECORDS OF A GIVEN DATA CODE SELECTION ----
             if(.not.lci) go to 410
    390      if(.not.lnm) go to 400
             write(9,340) dy(l),hr(l),rmn(l),lin(l),sp(l),isn(l),(idd(l,k),k=1,
             6),iu(l),(s(l,k),k=1,20),ic(l),im(l),ilt(l),iln(l)
             go to 410
    400      write(9,360) dy(l),hr(l),rmn(l),lin(l),sp(l),isn(l),(idd(l,k),k=1,
             6),iu(l),(s(l,k),k=1,20),ic(l),im(l)
    410      continue
             if(jm.eq.16) jm=1
             go to 100
    420      m=0
    430      no=no+1
             ni(no)=j
             if(.not.lci) go to 460
    440      if(.not.lnm) go to 450
             write(9,340) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
             6),iu(j),(s(j,l),l=1,20),ic(j),im(j),ilt(j),iln(j)
             go to 460
    450      write(9,360) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
             6),iu(j),(s(j,l),l=1,20),ic(j),im(j)
    460      if(m.ge.1) go to 20
    470      lsi=.true.
    480      lav=.true.
    490      return
             end
```

```
COMPILATION LISTING OF SEGMENT thsort
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1634.5 pst Tue
    Options: table symbols

             subroutine thsort
             logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm,lout
             real idt
             double precision dble,sv,svk
             character*4 isn,rb
             character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,shp1,shp2
             integer cod,dy,hi,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nsl,s,sp
             common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
             s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
             ,im(2400)
             common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
             nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
             ,lie,lpout,lnm,lout
c
c ---- IF SORTING ON TIME, BRANCH TO 1 ----
             if(.not.lhs) go to 1
c ---- IF ORDER OF INPUT DATA SET WAS CONTAINED IN DATA LINE INDEX VARIABLE,
c          BRANCH TO 16 ----
             if(lsi) go to 16
             go to 12
1            yrk=s(1,14)
             yrb=(yrk-60)*400
c
c ---- ASSIGN VALUES IN DECIMAL DAYS TO EACH RECORD ----
             do 10 i=1,n
             if(s(i,14).eq.yrk) go to 7
             yrk=s(i,14)
             yrb=(yrk-60)*400
7            sv(i)=dble(yrb)
             sv(i)=sv(i)+dy(i)+(hr(i)+(rmn(i)*.016667))*.041667
 10          continue
c ---- INITIALIZE OR RESET DATA LINE INDEX VARIABLE, NI ----
 12          do 14 i=1,n
 14          ni(i)=i
 16          nem1=n-1
c
c ---- SORT RECORDS ----
             do 40 i=1,nem1
             k=i
             svk=sv(k+1)
             ind=ni(k+1)
 20          if(svk.ge.sv(k)) go to 30
             sv(k+1)=sv(k)
             ni(k+1)=ni(k)
             k=k-1
             if(k.ge.1) go to 20
 30          if(k.eq.i) go to 40
             sv(k+1)=svk
             ni(k+1)=ind
 40          continue
c
c ---- LCI = F => NO OUTPUT.  ONLY DATA LINE INDEX VARIABLE PASSED TO OTHER
c           SUBROUTINES ----
             if(.not.lci) go to 100
             if(lav) go to 70
```

```
c ---- OUTPUT WITH NO ADDITIONAL VARIABLES ----
          do 60 i=1,n
          j=ni(i)
          write(9,50)  dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
          6),iu(j),(s(j,l),l=1,15)
  50      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
          t37,4a1,t66,a1,4i1,t73,a4,i2,a2)
  60      continue
          go to 100
c ---- OUTPUT WITH ADDITIONAL VARIABLES ----
  70      do 90 i=1,n
          j=ni(i)
          if(lnm) go to 82
          write(9,80)  dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
          6),iu(j),(s(j,l),l=1,20),ic(j),im(j)
  80      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
          t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3)
          go to 90
c ---- OUTPUT WITH ADDITIONAL VARIABLES PLUS NAVIGATION ----
  82      write(9,84)  dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
          6),iu(j),(s(j,l),l=1,20),ic(j),im(j),ilt(j),iln(j)
  84      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
          t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
  90      continue
  100     return
          end
```

```
COMPILATION LISTING OF SEGMENT selsch
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1849.0 pst Tue
    Options: table symbols

            subroutine selsch (segout)
            logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm,lout
            real idt
            double precision sv,svk
            character*4 isn,rb
            character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,
            shp1,shp2
            character*32 segout
            integer cod,dy,hi,hir,hm,hr,ic,it(2400),ik(1000),im,ios,iu,l1,l2,l3,
            l4,n,ni,nsl,s,sp
            common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
            s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400),
            im(2400)
            common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
            nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
            ,lie,lpout,lnm,lout
            common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
            shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
            med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
            l4(158)
c
            kf=0
            np=0
            nr=0
            kc=1
            kd=1
5           nei=0
c ---- NSL CONTAINS THE NUMBER OF SEARCH LEVELS IN EACH SEARCH SSEQUENCE ----
            kf=kf+nsl(kc)
c ---- INDEX OF 'S' VARIABLE TO BE SEARCHED ----
            nd=ios(kd)
c ---- KS = VALUE OF 'S' VARIABLE TO BE SEARCHED ----
            ng=ks(kd)
c
c
c ---- PERFORMS SEARCH OF INPUT CDI RECORDS FOR THE 'S' VALUES TO BE SEARCHED
c           FOR AT EACH SEARCH LEVEL OF A SEARCH SEQUENCE ----
            do 60 i=1,n
            j=i
            if(lsi) j=ni(i)
c ---- FIRST SEARCH LEVEL OF A SEARCH SEQUENCE ----
            if(s(j,nd).ne.ng) go to 60
c ---- NEI COUNTS FINDS OF 'S' VALUE BEING SEARCHED FOR IN THE FIRST SEARCH
c         LEVEL OF A SEARCH SEQUENCE ----
            nei=nei+1
c ---- IK IS INDEX VARIABLE THAT STORES DATA LINE NUMBER OF FINDS.  USED IN
c          STATEMENT 90 IF THE FIRST SEARCH LEVEL IN THE NEXT SEARCH SEQUENCE
c          IS THE SAME ----
            ik(nei)=j
            if(nsl(kc).eq.1) go to 40
            ke=kd+1
c
c ---- FURTHER SEARCH LEVELS OF A SEARCH SEQUENCE ----
            do 20 nk=ke,kf
            nf=ios(nk)
```

```
           if(s(j,nf).ne.ks(nk)) go to 60
20         continue
c ---- NR IS COUNT OF ALL RECORDS WHICH ARE RECOVERED IN ALL SEARCH SEQUENCES ----
40         nr=nr+1
           it(nr)=j
60         continue
c
c ---- ICC IS COUNT OF RECORD FINDS FOR SEARCH SEQUENCE JUST CONDUCTED ----
70         icc=nr-np
           write(6,71) icc,kc
71         format("0",i4," RECORDS FOUND FOR SEARCH SEQUENCE ",i2)
           np=nr
c ---- KC IS COUNT OF CURRENT SEARCH SEQUENCE BEING PERFORMED ----
           kc=kc+1
c ---- NSS = NUMBER OF SEARCH SEQUENCES TO BE PERFORMED ----
           if(kc.gt.nss) go to 140
c ---- KD COUNTS THE TOTAL NUMBER OF INDIVIDUAL SEARCHES THAT HAVE BEEN
c         PERFORMED.  IT KEEPS TRACK OF THE 'S' VALUE (KS) CURRENTLY BEING
c         SEARCHED FOR ----
           kd=kd+nsl(kd)
           if(ks(kd).eq.ng) go to 80
           go to 5
80         kf=kf+nsl(kc)
           ke=kd+1
c ---- IF THE 'S' VARIABLE IN THE FIRST SEARCH LEVEL OF THE NEXT SEARCH
c         SEQUENCE IS THE SAME AS THE 'S' VARIABLE IN THE FIRST SEARCH LEVEL
c         OF THE PREVIOUS SEARCH SEQUENCE, THE PROGRAM SECTION BEGINNING WITH
c         STATEMENT 90 USES THE STORAGE OF THIS VARIABLE IN IK TO PREVENT DUPLI-
c         CATE EFFORT ----
90         do 120 i=1,nei
           if(nsl(kc).eq.1) go to 110
           j=ik(i)
c ---- FURTHER SEARCHES PERFORMED ON RECORDS CONTAINING AN 'S' VALUE PRE-
c         VIOUSLY SELECTED OUT OF THE CDI DATA SET AND STORED IN IK ----
           do 100 nk=ke,kf
           nf=ios(nk)
           if(s(j,nf).ne.ks(nk)) go to 120
100        continue
110        nr=nr+1
           it(nr)=j
120        continue
           go to 70
140        if(lci) go to 150
c
c ---- SETS UP CONDITIONS NECESSARY TO OUTPUT SEARCHED RECORDS USING SUB-
c         ROUTINE FINPRN ----
           nrec=n
           n=nr
           call io ("attach","file42","vfile_",segout)
           call io ("open","file42","so")
           do 144 i=1,n
144        ni(i)=it(i)
           nlpp=60
           ncpds=1
           lph=.false.
           lhs=.false.
           call finprn
           call io ("close","file42")
           call io ("detach","file42")
           n=nrec
```

```
         write(6,146)
146      format("LSI SET = TO FALSE. ORDER OF DATA STORED IN INDEX VARIAB",%
         "LE IS KAPUT.")
         lsi=.false.
         return
c
c --- WRITES OUT SEARCHED RECORDS IN SAME FORMAT AS INPUT CDI DATA SET ----
150      if(lav) go to 162
         do 160 i=1,nr
         j=it(i)
154      write(9,155) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
         6),iu(j),(s(j,l),l=1,15)
155      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
         t37,4a1,t66,a1,4i1,t73,a4,i2,a2)
160      continue
         go to 176
162      do 172 i=1,nr
         j=it(i)
         if(lnm) go to 166
         write(9,164) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
         6),iu(j),(s(j,l),l=1,20),ic(j),im(j)
164      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
         t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3)
         go to 172
166      write(9,168) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
         6),iu(j),(s(j,l),l=1,20),ic(j),im(j),ilt(j),iln(j)
168      format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
         t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
172      continue
         endfile 09
176      call io ("close","file09")
         call io ("detach","file09")
         return
         end
```

```
COMPILATION LISTING OF SEGMENT navmer
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1636.3 pst Tue
    Options: table symbols
```

```
          subroutine navmer (lmarc,lmgvx,kfor)
          logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm
          ,lout,lssln,lndn,lsmn,lpas,lmgvx,lgap,lmarc
          real idt
          double precision dam1,dam2,dlat1,dlat2,dlon1,dlon2,dsdlat,dsdlon,
          dcse,dspd,dumya8,sv,svk
          character*4 isn,rb
          character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,
          shp1,shp2
          integer cod,dy,hi,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nsl,s,sp
          common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
          s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
          ,im(2400)
          common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
          nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
          ,lie,lpout,lnm,lout
          common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
          shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
          med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
          l4(158)
c
          lndn=.false.
          lsmn=.false.
          lpas=.false.
          lgap=.false.
          numer=0
          ner=0
c
c ---- READ IN NAVIGATIONAL RECORDS ----
          if(lmgvx) go to 20
          if(lmarc) go to 25
          go to (2,3,4),kfor
2         read(10,10)nvd1,nvh1,rmnv1,rltn1,rlnn1
          read(10,10)nvd2,nvh2,rmnv2,rltn2,rlnn2
          go to 30
3         read(10,11)nvd1,nvh1,rmnv1,rltn1,rlnn1
          read(10,11)nvd2,nvh2,rmnv2,rltn2,rlnn2
          go to 30
4         read(10,12)nvd1,nvh1,rmnv1,rltn1,rlnn1
          read(10,12)nvd2,nvh2,rmnv2,rltn2,rlnn2
10        format(i3,1x,i2,f4.1,43x,f9.5,f10.5)
11        format(i3,1x,i2,f4.1,43x,f8.5,f10.5)
12        format(i3,1x,i2,f4.1,44x,f7.5,1x,f9.5)
          go to 30
c ---- READ IN NAVIGATION RECORDS FROM MAGNAVOX TAPE ----
20        read(10) nvd1,nvh1,rmnv1,rltn1,rlnn1
          read(10) nvd2,nvh2,rmnv2,rltn2,rlnn2
          go to 30
c ---- READ IN NAVIGATION RECORDS FROM MARCONI TAPE ----
25        read(10) nvd1,nvh1,dam1,iline,ishot,itype,dlat1,dsdlat,dlon1,
          dsdlon,dcse,dspd,dumya8
          read(10) nvd2,nvh2,dam2,iline,ishot,itype,dlat2,dsdlat,dlon2,
          dsdlon,dcse,dspd,dumya8
          rmnv1=sngl(dam1)
          rmnv2=sngl(dam2)
```

```
            rltn1=sngl(dlat1)
            rlnn1=sngl(dlon1)
            rltn2=sngl(dlat2)
            rlnn2=sngl(dlon2)
c
c
c ---- MERGES NAVIGATION WITH SCIENTIFIC CDI DATA RECORDS.  SKIPS MERGE
c        SECTION FOR NON-SCIENTIFIC RECORDS. ----
30          do 310 i=1,n
            j=i
            if(lsi) j=ni(i)
            if(lndn) go to 270
            jm=im(j)
            mg=hm(jm)
            if(mg.lt.6) go to 40
            if(mg.gt.7) go to 50
c ---- SENDS RECORDS WITH MEDIA CODES, CRD AND DIG AROUND MERGING SECTION ----
            go to 270
c ---- MED(13) = MAP ----
40          if(s(j,1).eq.med(13)) go to 270
            go to 60
c ---- SENDS RECORD WITH MEDIA CODES PER, EQP AND DATA CODES OF CRUZ AND PORT
c        AROUND MERGING SECTION ----
50          if(mg.gt.10) go to 270
c ---- LNM = TRUE => CDI DATA SET ALREADY PARTIALLY MERGED WITH NAVIGATION ----
60          if(.not.lnm) go to 70
c ---- NAV MERGED RECORDS SENT AROUND MERGING SECTION ----
            if(ilt(j).ne.0) go to 290
c ---- LPAS = TRUE => HAVE ENCOUNTERED A NAV RECORD TIME = OR < A CDI RECORD
c        TIME ----
70          if(lpas) go to 130
c ---- COMPARES CDI RECORD TIME TO NAV RECORD TIMES ----
            tdif1=1440*(dy(j)-nvd1)+60*(hr(j)-nvh1)+rmn(j)-rmnv1
c ---- TDIF1 < 0 => CDI RECORD TIME < NAV RECORD TIME ----
            if(tdif1) 80,110,120
80          ner=ner+1
            if(ner.gt.5) go to 270
            write(6,90) dy(j),hr(j),rmn(j),nvd1,nvh1,rmnv1
90          format(" ***WARNING: RECORD IS EARLIER THAN 1ST NAV. RECORD."
            /"     CDI DAY/TIME OF  ",i3,"/",i2,f4.1,", NAV DAY/TIME OF ",
            i3,"/",i2,f4.1)
            if(ner.eq.5) write(6,100)
100         format("0  MESSAGES SUPPRESSED. CONTINUE CDI/NAV COMPARISONS.")
            go to 270
c ---- CDI AND 1ST NAV RECORD HAVE SAME TIME, CDI NAV SET = TO THAT OF 1ST
c        NAV RECORD ----
110         rlt=rltn1
            rln=rlnn1
            go to 280
120         lpas=.true.
c ---- COMPARE CDI RECORD TIME WITH NEXT NAV RECORD TIME ----
130         tdif2=1440*(nvd2-dy(j))+60*(nvh2-hr(j))+rmnv2-rmn(j)
            if(tdif2) 230,220,140
c ---- LSMN = TRUE => SAME NAVIGATION AS PREVIOUS PASS ----
140         if(lsmn) go to 190
c ---- CALCULATES TIME DIFFERENCE BETWEEN SUCCESSIVE NAV RECORDS ----
150         tdifn=1440*(nvd2-nvd1)+60*(nvh2-nvh1)+rmnv2-rmnv1
            dlt=rltn2-rltn1
            dln=rlnn2-rlnn1
c ---- LSSLN = TRUE => LONGITUDES OF SUCCESSIVE NAV RECORDS HAVE SAME SIGNS ----
```

```
            lssln=.true.
            if(rlnn1*rlnn2.gt.0) go to 160
c ---- ADJUST NAV DIFFERENCES FOR NAV RECORDS W/DIFFERENT SIGNS ----
            if(dln.gt.180.) dln=dln-360.
            if(dln.lt.-180.) dln=dln+360.
            lssln=.false.
160         if(tdifn.lt.361) go to 200
c ---- LGAP = TRUE =OF TIME GAP BETWEEN CURRENT SUCCESSIVE NAV RECORDS OF
c            > 3 HOURS ----
            lgap=.true.
170         if(tdif2.lt.121) go to 200
            tdif3=1440*(dy(j)-nvd1)+60*(hr(j)-nvh1)+rmn(j)-rmnv1
            if(tdif3.lt.121) go to 200
c ---- TIME GAP BETWEEN CDI RECORD AND EACH SURROUNDING NAV RECORD > 2 HOURS ----
            numer=numer+1
            if(numer.gt.5) go to 185
            write(6,180) dy(j),hr(j),rmn(j)
180         format("0***ATTENTION: CDI RECORD OF DAY/TIME ",i3,"/",i2,f4.1,
            " IS > 2 HRS AWAY FROM NEAREST NAV RECORD, AND BRACKETING NAV"/
            "       RECORDS ARE > 6 HRS APART. THEREFORE, INTERPOLATION SKIPPE",%
            "D;"/"      AND LAT/LONG FOR CDI RECORD WILL BE SET TO ZERO.")
            if(numer.eq.5) write (6,100)
185         lsmn=.true.
            go to 270
190         if(lgap) go to 170
c
c ---- INTERPOLATES CDI RECORD NAVIGATION ----
200         tr=tdif2/tdifn
            rlt=rltn2-tr*dlt
            rln=rlnn2-tr*dln
            if(lssln) go to 210
            if(rln.gt.180.) rln=rln-360.
            if(rln.lt.180.) rln=rln+360.
210         lsmn=.true.
            go to 280
220         rlt=rltn2
            rln=rlnn2
            go to 280
c
c ---- MOVE TIME AND POSITION FOR 2ND NAV  RECORD INTO TIME AND NAV VARIABLES
c            FOR 1ST NAV RECORD. ----
230         nvd1=nvd2
            nvh1=nvh2
            rmnv1=rmnv2
            rltn1=rltn2
            rlnn1=rlnn2
            lsmn=.false.
            lgap=.false.
c
c ---- READ NEW 2ND NAV RECORD AND BRANCH TO STATEMENT COMPARING IT WITH
c            CURRENT CDI RECORD ----
            if(lmgvx) go to 240
            if(lmarc) go to 250
            go to (233,234,235),kfor
233         read(10,10,end=260) nvd2,nvh2,rmnv2,rltn2,rlnn2
            go to 130
234         read(10,11,end=260)nvd2,nvh2,rmnv2,rltn2,rlnn2
            go to 130
235         read(10,11,end=260)nvd2,nvh2,rmnv2,rltn2,rlnn2
            go to 130
```

```
240   read(10,end=260) nvd2,nvh2,rmnv2,rltn2,rlnn2
      go to 130
250   read(10,end=260) nvd2,nvh2,dam2,iline,ishot,itype,dlat2,dsdlat,
      dlon2,dsdlon,dcse,dspd,dumya8
      rmnv2=sngl(dam2)
      rltn2=sngl(dlat2)
      rlnn2=sngl(dlon2)
      go to 130
c
260   m=j
c ---- LNDN = TRUE IMPLIES END OF NAV DATA SET REACHED ----
      lndn=.true.
270   if(lnm) go to 290
      ilt(j)=0
      iln(j)=0
      go to 290
280   ilt(j)=rlt*100000.
      iln(j)=rln*100000.
290   if(.not.lci) go to 310
c ---- WRITES MERGED RECORD TO FILE09 IF LCI = TRUE ----
      write(9,300) dy(j),hr(j),rmn(j),lin(j),sp(j),isn(j),(idd(j,l),l=1,
      6),iu(j),(s(j,l),l=1,20),ic(j),im(j),ilt(j),iln(j)
300   format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
      t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
310   continue
c
      if(lndn) go to 330
      write(6,320)
320   format("0 END OF CDI DATASET REACHED. RETURN TO COMMAND PROMPT.")
      return
330   write(6,340) dy(m),hr(m),rmn(m)
340   format("0**** POSSIBLE ERROR: END OF NAVIGATION DATASET REACHED ",%
      "WHILE"/" LOOKING FOR NAV RECORD WITH DAY/TIME OF ",i3,"/",i2,f4.1)
      return
      end
```

```
COMPILATION LISTING OF SEGMENT finprn
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1637.0 pst Tue
     Options: table symbols

      subroutine finprn
      logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm
      ,lout,lac,lhi,lad,lab
      real idt
      double precision sv,svk
      character*4 isn,rb,stn(2)
      character*8 cs1,cs2,date,dout,dsh1,dsh2,fun(2),ibk,ids,ist,mot1,
      mot2,mout,new,newl,shp1,shp2
      integer cod,dy,hi,hid,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nsl,
      s,sp,yrk
      common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
      s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
      ,im(2400)
      common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
      nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
      ,lie,lpout,lnm,lout
      common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
      shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
      med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
      l4(158)
      equivalence (ids,stn(1))
      data fun/"ARRIVE","LEAVE"/
c
c ---- INITIALIZE VARIABLES ----
      stn(1)=rb
      rlt=0.0
      rln=0.0
      jn=0
      mh=20
      nro=0
      nnn=0
      nlo=0
      yrk=0
      idk=ib
      ln=17
c ---- NUMBER OF LINES PER PAGE.  60 FOR HONEYWELL LINE-PRINTER ----
      nlpp=60
      lad=.true.
c
c
c
c ---- LOOP WHICH PROCESSES AND WRITES OUT SUCCESSIVE RECORDS OF A CDI
c         DATA SET ----
      do 1020 i=1,n
      j=i
      if(lsi)j=ni(i)
      ids=ibk
      nn=3
c
c ---- TRANSLATES ST/ED/ON/OFF INDICATOR IN S(4-7) TO NUMERIC VALUE OF
c         1,2,3, OR 4 AND STORES IN ISTA ----
      do 10 m=1,4
      nn=nn+1
      if(s(j,nn).eq.ix) go to 20
 10   continue
```

```
          m=0
c
c ---- ASSIGNS 'START', 'END', 'ON', 'OFF', 'RECOVERY' OR 'NO RECO' TO RECORD
c        OUTPUT VARIABLE IDS ----
          if(s(j,8).eq.ir) ids=ist(5)
          if(s(j,8).eq.in) ids=ist(6)
          go to 30
20        ids=ist(m)
c
c ---- ASSIGNS NUMBERS LOCATING MEDIA AND DATA CODES IN MED AND COD VARIABLES
c        TO JM AND JC ----
30        jm=im(j)
          jc=ic(j)
c ---- ASSIGNS SEQUENCE NUMBERS TO CDI RECORD BASED ON JM ----
          mg=hm(jm)
c
c ---- HIERARCHAL SORT INPUT TO 50 ----
          if(lhs) go to 50
c ---- IF CURRENT RECORD'S CRUISE LOCATOR NOT = TO PREVIOUS RECORD'S CRUISE
c        LOCATOR, SHUNTS TO 430 AND STARTS NEW PAGE ----
40        if(s(j,13).ne.idk.or.s(j,14).ne.yrk) go to 430
c ---- NLO COUNTS LINES OF OUTPUT.  RESET TO ZERO WHEN IT REACHES 60.
c        IF NLO < 60, SENDS RECORDS TO OUPUT SECTION ----
          if(nlo.lt.nlpp) go to 540
          go to 150
c
c ---- MG NOT = MH => CURRRENT RECORD IS START OF NEW DATA GROUP ----
50        if(mg.ne.mh) go to 80
c ---- RECORDS WITH MEDIA CODES OF EQP AND PER AND DATA CODES OF PORT AND
c        CRUZ SHUNTED TO 70 AND 40 WITHOUT GOING THRU PAGE SKIP TEST ----
          if(mg.lt.11) go to 60
          go to 70
c ---- SEPARATES RECORDS WHICH HAVE SAME MEDIA CODE BUT SLIGHTLY DIFFERENT
c        HIERARCHAL CODES.  EG. STAT AND TRAK RECORDS ----
60        if(hir.ne.s(j,20)) go to 80
70        lhi=.false.
          go to 40
c
c ---- STATEMENTS 80 THRU 140 DEAL WITH THE FIRST RECORD OF EACH NEW DATA
c        GROUP AND DETERMINES WHETHER THE GROUP SHOULD HAVE AN EXTENDED DATA
c        HEADER OR NOT OR WHETHER IT SHOULD START AT THE TOP OF A NEW PAGE ----
80        lhi=.true.
          hir=s(j,20)
c ---- LAC = TRUE => DATA HEADER EXTENDED TO INCLUDE 'DATA DESCRIPTION/
c        COMMENTS OR OBSERVATIONS ----
          lac=.false.
c ---- SENDS VESSEL AND GEOPHYSICAL RECORDS TO 90 ----
          if(hir.lt.300000) go to 90
          go to 100
c ---- MH = 11 => PREVIOUS RECORD WAS PORT OR CRUZ RECORD. LAD = F => PREVI-
c        OUS RECORD DID NOT HAVE A 'DATA DESCRIPTION/COMMENTS OR OBSERVATION
c        SECTION ----
90        if(mh.eq.11) lad=.false.
c ---- ALL VESSEL AND GEOPHYSICAL RECORDS EXCEPT STATION, OBSERVATION AND
c        DEPLOYMENT RECORDS RETAIN 'FALSE' VALUE FOR LAC ----
          if(jm.eq.8.or.jm.eq.12) lac=.true.
          if(hir.ne.108110) go to 110
c ---- LAC SET = TO TRUE FOR ALL GEOLOGIC, HYDROGRAPHIC, NAV STATION RECORDS.
c        IMPLIES THAT DATA HEADER EXTENDED AND RECORDS WRITTEN OUT WITH
c        'DATA DESCRIPTION/COMMENTS OR OBSERVATIONS' ----
```

```
100       lac=.true.
110       if(s(j,13).ne.idk.or.s(j,14).ne.yrk) go to 430
c ---- WHEN NLO = NLPP, PRINTER AUTOMATICALLY POSITIONS AT TOP OF NEW PAGE ----
          if(nlo.lt.nlpp) go to 120
          lad=lac
          go to 150
c ---- IF CURRENT RECORD DIFFERS FROM PREVIOUS RECORD IN HAVING ADDITIONAL
c        COMMENTS SECTION, A NEW PAGE WILL BE STARTED ----
120       if(lac.eq.lad) go to 550
          lad=lac
c ---- POSITION PRINTER AT START OF NEW PAGE ----
130       write(42,140)
140       format("1")
          go to 160
150       write(42,520)
c ---- PAGE LINE COUNT RESET TO ZERO ----
160       nlo=0
170       if(.not.lph) go to 180
          go to 490
180       if(lhs) go to 290
c
c ---- WRITES OUT DATA HEADERS FOR EACH PAGE OF TIME SORTED OUTPUT WHEN PAGE
c        HEADERS NOT DESIRED ----
          write(42,190)dsh1,dsh1,dsh1,dsh2,dsh2
190       format("+",14a8,"-")
          write(42,200)dsh2
200       format("+",t115,a8,"----")
          write(42,210)
210       format(" ",t31,"CRUISE/DATA   SAMPLE/",t62,"PERSONNEL,PORTS,EQUI",%
          "PMENT WATER")
          write(42,220)
220       format("+",t118,"CRUISE")
          write(42,230)
230       format(" JUL. TIME  DATA TYPE/SYSTEM INFO OR RECOR- SEQNCE   DAT",%
          "A     DESCRIPTION  OR:            DEPTH  LATITUDE  LONGITUDE")
          write(42,240)
240       format("+",t119,"LOCATOR")
          write(42,250)
250       format(" DAY  (GMT)   OR EQUIPMENT    DING MEDIUM   NUMBER  STAT",%
          "US    LINE#    STA./SHOT PT.# UNCOR. DEG  MIN   DEG  MIN")
          write(42,260)
260       format("+",t118,"ID -YR-AR")
          write(42,270)
270       format(" --- ------ ---------------- ------------- ------ -----",%
          "---- -------- ---------------- ------ --- ----- ---- -----")
          write(42,280) dsh2
280       format("+",t117,a8,"--")
          write(42,520)
          nlo=nlo+6
          go to 540
c
c ---- WRITES OUT DATA HEADERS FOR EACH PAGE OF HIERARCHALLY SORTED OUTPUT
c        WHEN PAGE HEADERS NOT DESIRED ----
290       write(42,300)dsh1,dsh1,dsh2,dsh2,dsh2
300       format("+",11a8,"----")
          if(lac) write(42,310)dsh2,dsh2,dsh2
310       format("+",t108,3a8,"-")
          write(42,320)dsh2
320       format("+",t96,a8,"--")
          write(42,330)
```

```
330       format(" ",t14,"CRUISE/DATA INFO",t33,"DATA",t41,"PERSONNEL,PORT",%
          "S,EQUIPMENT WATER")
          write(42,340)
340       format("+",t98,"CRUISE")
          write(42,350)
350       format(" JUL. TIME    RECORD. SEQNCE   STATUS/  DESCRIPTION  OR:",
          t67,"DEPTH  LATITUDE  LONGITUDE")
          if(lac) write(42,360)
360       format("+",t108,"DATA DESCRIPTION,COMMENTS")
          write(42,370)
370       format("+",t99,"LOCATOR")
          write(42,380)
380       format(" DAY  (GMT)   MEDIUM  NUMBER  INSTITUTE  LINE#   STA./SH",%
          "OT PT.#   UNCOR. DEG   MIN    DEG  MIN")
          if(lac) write(42,390)
390       format("+",t113,"OR OBSERVATIONS")
          write(42,400)
400       format("+",t98,"ID -YR-AR")
          write(42,410)
410       format(" --- ------  -------- ------- --------- ------- --------",%
          "--------- ------ --- ----- ---- -----")
          if(lac) write(42,310)dsh2,dsh2,dsh2
          write(42,420)
420       format("+",t96,"---- -- --")
          write(42,520)
          nlo=nlo+6
          go to 560
c
c ---- TESTS RECORD'S CRUISE ID AGAINST CRUISE ID'S ENTERED IN CDIMAIN TO
c        MAKE SURE THEY ARE THE SAME ----
430       if(mh.ne.20) write(42,140)
c ---- STORES CURRENT RECORD'S CRUISE LOCATOR (ID & YEAR) IN IDK AND YRK TO
c        TEST AGAINST CRUISE LOCATOR OF NEXT RECORD ----
440       idk=s(j,13)
          yrk=s(j,14)
          nlo=0
          do 450 nz=1,ncpds
          if(s(j,13).eq.ida(nz)) go to 480
450       continue
c ---- GIVES OPTION OF ABORTING OR CONTINUING IF OFF-THE-WALL CRUISE ID IS
c        ENCOUNTERED ----
          write(6,460) s(j,13)
460       format("0****ERROR: NO SHIP AND CHIEF SCIENTIST DATA FOR CRUISE ",%
          "ID ",a4/"      ABORT?? IF TRUE, PROGRAM RETURNED TO COMMAND PRO",%
          "MPT. IF f, INDEX TITLE AND HEADER NOT PRINTED. (t OR f) (L1)")
          read(5,470) lab
470       format(l1)
          if(lab) return
          lph=.false.
          go to 180
c ---- WRITES OUT TITLE FOR CDI FINAL-PRINT OUTPUT ----
480       if((nz-nnn).gt.0) call title (shp1,shp2,cs1,cs2,date,s,lhs,lnm,nz,
          j)
          nnn=nz
          if(.not.lph) go to 180
c
c ---- WRITES OUT PAGE HEADERS FOR BOTH TIME AND HIERARCHALLY SORTED OUTPUT ----
490       write(42,500) shp1(nz),shp2(nz),(s(j,jj),jj=13,15),cs1(nz),cs2(nz)
500       format("+",t19,"SHIP:  ",2a8,t60,"CRUISE LOCATOR:  ",a4,"-",i2,"-"
          ,a2/" ",t77," ID -YR-AREA"/" ",t19,"CHIEF"/" ",t19,"SCIENTIST:  ",
          2a8)
          write(42,520)
510       if(lhs) go to 530
c
c ---- WRITES OUT DATA HEADERS FOR EACH PAGEE OF TIMESORT OUTPUT WHEN PAGE
c        HEADERS DESIRED ----
          write(42,190)dsh1,dsh1,dsh1,dsh2,dsh2
          write(42,210)
          write(42,230)
          write(42,250)
          write(42,270)
          write(42,520)
520       format(" ")
          nlo=nlo+10
          go to 830
c
c ---- WRITES OUT DATA HEADERS FOR EACH PAGE OF HIERARCHALLY SORTED OUTPUT
c        WHEN PAGE HEADERS DESIRED ----
530       write(42,300)dsh1,dsh1,dsh2,dsh2,dsh2
          if(lac) write(42,310)dsh2,dsh2,dsh2
          write(42,330)
          write(42,350)
          if(lac) write(42,360)
          write(42,380)
          if(lac) write(42,390)
          write(42,410)
          if(lac) write(42,310)dsh2,dsh2,dsh2
          write(42,520)
          nlo=nlo+10
          go to 560
c
c ---- IF LHS = F, RECORD SENT TO TIMESORT SECTION ----
540       if(.not.lhs) go to 830
c
c ---- ROUTES RECORDS OF CURRENT DATA GROUP TO LINE OUT-PUT SECTION ----
          if(jn.gt.0) go to 680
c ---- K AND KK LOCATE EXPANDED DATA DESCRIPTION IN 'DOUT' ----
          k=2*jc-1
          kk=k+1
c ---- CRUZ, PORT, EQP, AND PER RECORDS GO TO 760 ----
          go to 760
c
c ---- IF PAGE LINECOUNT IS WITHIN 7 LINES OF A FULL PAGE, A NEW PAGE IS
c        STARTED ----
550       if((nlo+7).gt.nlpp) go to 130
c
c
c ---- THIS SECTION WRITES OUT A DATA DESCRIPTOR AT THE HEAD OF EACH NEW
c        DATA GROUP ----
          write(42,520)
          nlo=nlo+1
560       nlo=nlo+3
c ---- K AND KK LOCATE EXPANDED DATA DESCRIPTION IN 'DOUT' ----
          k=2*jc-1
          kk=k+1
c ---- L AND LL LOCATE EXPANDED MEDIA DESCRIPTION IN 'MOUT' ----
          l=2*jm-1
          ll=l+1
          mh=mg
c ---- ASSIGNS NEW SEQUENCE NUMBER TO A RECORD BASED ON MG AND HENSE, JM ----
```

```
          it=mg-9
c ---- IT < OR = 0 => RECORD IS NOT A CRUZ, PORT, TRAK, STAT, PER OR EQP
c        RECORD ----
          if(it.le.0) go to 635
          go to (620,570,590,590),it
c ---- WRITES DATA DESCRIPTOR FOR CRUZ AND PORT RECORD GROUP ----
570       write(42,580)
580       format("0",t16,"CRUISE DATES AND PORT STOPS")
          go to 750
c ---- WRITES DATA DESCRIPTOR FOR PER AND EQP RECORD GROUPS ----
590       write(42,600) mout(l),mout(ll)
600       format("0",t16,a8,a6)
          go to 750
c ---- WRITES DATA DESCRIPTORS FOR TRAK AND STAT RECORD GROUPS ----
620       jn=jc-25
          write(42,630) dout(k),dout(kk)
630       format("0",t16,a8,a6,"S")
          go to 660
c ---- SETS CONDITIONS FOR WRITING DATA DESCRIPTORS FOR NON-CRUZ, PORT,
c        TRAK, ETC., RECORD GROUPS ----
635       jn=jm-3
c ---- WRITES DATA DESCRIPTORS FOR NON-CRUZ, PORT, ETC., RECORD GROUPS ----
640       write(42,650) dout(k),dout(kk),mout(l),mout(ll)
650       format("0",t16,2a8,1x,a8,a6,"S")
c ---- WRITES '(CONTINUED)' JUST AFTER DATA DESCRIPTOR IF SAME DATA GROU
c        CONTINUED ON NEW PAGE ----
660       if(.not.lhi) write(42,670)
670       format("+",t49,"(CONTINUED)")
c
c
c ---- TRANSLATES HIERARCHALLY SORTED CDI RECORD VARIABLES TO MORE READA
c        OUTPUT IN FINAL-PRINT FORMAT ----
          write(42,520)
680       newl=new(jn)
c ---- JM = 7 => SYS RECORDS ----
          if(jm.eq.7) go to 700
690       if(m.gt.2) newl=ibk
c ---- WRITES THE MEDIA OR GENERAL SAMPLE TYPE, THE SEQUENCE NUMBER (ROL
c        SAMPLE, ETC.), AND THE LINE AND SHOT POINT OR STATION NUMBER --
700       write(42,710) newl,s(j,3),isn(j),lin(j),sp(j)
710       format(" ",t14,a8,1x,a1,a4,t41,"L# ",a4,3x,"STN/SP# ",a4)
c ---- SENDS SAMPLE RECORDS TO 715 ----
          if(jm.eq.4) go to 715
c ---- SENDS ALL RECORDS REQUIRING 'ADDITIONAL COMMENTS' TO 720 ----
          if(lac) go to 720
c ---- SENDS ALL OTHER RECORDS TO 790 ----
          go to 790
715       if(m.ne.2) go to 720
c ---- PUTS 'END NO R' IN OUTPUT VARIABLE IDS FOR THE 'END' RECORDS OF
c        CONTINUOUS UNDERWAY SAMPLES, EG. DREDGES ----
          ids=ist(8)
          ln=14
          if(s(j,8).ne.ir) go to 730
c ---- PUTS 'END RECO' IN IDS FOR RECOVERED CONTINUOUS UNDERWAY SAMPLES ----
          ids=ist(7)
          ln=16
          go to 730
720       ln=17
c ---- WRITES WATER DEPTH, UNITS AND DATA COMMENTS FIELD ----
730       write(42,740) idas(ln),(s(j,jj),jj=9,12),iu(j),(idd(j,jj),jj=1,6)
```

```
740       format("+",t39,a1,t67,4i1,1x,a1,t108,6a4)
          go to 790
c
c ---- SETS UP CONDITIONS FOR WRITING PORT,RTIONS OF DATA LINE FOR CRUZ, PORT,
c       EQP AND PERS RECORDS ----
c ---- WRITES '(CONTINUED)' IF LHI = FALSE ----
750       if(.not.lhi) write(42,670)
          write(42,520)
760       jn=0
          if(mg.ne.12) go to 770
c ---- TRANSFERS PERSON'S INSTITUTE CODE, EG. UGS, FROM ISN TO OUTPUT
c         VARIABLE ----
          stn(1)=isn(j)
          stn(2)=rb
c ----PLACES LITERAL 'ON' OR 'OFF' IN STN(2) FOR PERSONNEL RECORDS ----
          if(m.gt.2) stn(2)=ist(m+13)
c ---- WRITES DATA-TYPE NAME AND DATA COMMENTS ----
770       write(42,780) dout(k),dout(kk),(idd(j,jj),jj=1,6)
780       format(" ",t14,2a8,t41,6a4)
          if(jc.ne.2) go to 790
c ---- PUTS LITERAL 'ARRIVE' OR 'LEAVE' IN OUTPUT VARIABLE IDS FOR PORT
c         STOP RECORDS ----
          ids=fun(1)
          if(m.eq.2) ids=fun(2)
c ---- WRITES DAY/TIME AND DATA STATUS ----
790       write(42,800) dy(j),hr(j),rmn(j),ids
800       format("+",i3,1x,i2,f4.1,t31,a8)
810       if(lph) go to 970
c ---- IF PAGE HEADER NOT DESIRED, WRITES CRUISE LOCATOR IN EACH DATA LINE ----
          write(42,820)s(j,13),s(j,14),s(j,15)
820       format("+",t96,a4,"-",i2,"-",a2)
          go to 970
c
c
c ---- THIS SECTION TRANSLATES TIMESORT CDI RECORD VARIABLES TO MORE READABLE
c         OUTPUT IN FINAL-PRINT FORMAT ----
c ---- JC EXPANDED TO K AND KK WHICH ARE USED TO LOCATE EXPANDED DATA
c         DESCRIPTIONS IN DATA STATEMENT 'DOUT' ----
830       k=2*jc-1
          kk=k+1
          mh=mg
c ---- WRITES DAY, TIME, SLAVE RECORDER, AND SEQUENCE/SAMPLE NUMBER ----
          write(42,840) dy(j),hr(j),rmn(j),s(j,3),isn(j)
840       format(" ",i3,1x,i2,f4.1,t45,a1,a4)
          if(mg.gt.10) go to 910
c ---- WRITES LINE OR SHOT POINT/STATION NUMBER ----
          write(42,850) lin(j),sp(j)
850       format("+",t62,"LN# ",a4,3x,"STN/SP# ",a4)
c ---- SHUNTS STATION RECORDS TO  860 ----
          if(k.eq.77) go to 860
c ---- SHUNTS GEOLOGIC/HYDROGRAPHIC RECORDS TO 860 ----
          if(s(j,20).gt.299999) go to 860
c ---- SHUNTS NON-SAMPLE RECORDS TO 880 ----
          if(mg.ne.8) go to 880
c ---- IF SAMPLE (DREDGE) IS A 'START', SHUNTS TO 890 ----
          if(m.eq.1) go to 890
c ---- ASSIGNS 'END NO R' TO OUTPUT VARIABLE IDS ----
          ids=ist(8)
c ---- IF SAMPLE IS A 'NO RECOVERY', SHUNTS TO 860 ----
          if(s(j,8).ne.ir) go to 860
```

```
c ---- ASSIGNS 'END RECO' TO OUTPUT VARIABLE IDS ----
           ids=ist(7)
c ---- WRITES DEPTH AND UNITS ----
860        write(42,870) (s(j,jj),jj=9,12),iu(j)
870        format("+",t88,4i1,a1)
c ---- SENDS NAV TRACKLINE AND STATION RECORDS TO 930 ----
880        if(mg.eq.10) go to 930
c ---- EXPANDS JM TO L AND LL WHICH LOCATE EXPANDED 'MEDIA' DESCRIPTIONS IN
c         DATA STATEMENT MOUT ----
           l=2*jm-1
           ll=l+1
           mot1=mout(l)
           mot2=mout(ll)
c ---- SENDS START/END RECORDS TO 890 ----
           if(m.lt.3) go to 890
           mot1=ibk
           mot2=ibk
c ---- WRITES DATA AND MEDIA DESCRIPTIONS AND STATUSRT/END/ON/OFF, ETC. STATUS
c          OF ALL RECORDS OTHER THAN NAV, PER ----
890        write(42,900) dout(k),dout(kk),mot1,mot2,ids
900        format("+",t13,2a8,1x,a8,a6,t52,a8)
           go to 950
c ---- WRITES 'COMMENTS/DESCRIPTION' FIELD FROM CDI LOGS ----
910        write(42,920) (idd(j,jj),jj=1,6)
920        format("+",t62,6a4)
c ---- SENDS NON-EQUIPMENT RECORDS TO 930 ----
           if(mg.ne.13) go to 930
           mot1=ibk
           mot2=ibk
           go to 890
c ---- WRITES DATA AND MEDIA DESCRIPTIONS AND START/END/ON/OFF STATUS FOR NAV
c         AND PER RECORDS ----
930        write(42,940) dout(k),dout(kk),ids
940        format("+",t30,a8,a6,t52,a8)
950        if(lph) go to 970
c ---- WRITES CRUISE LOCATOR FOR EACH LINE IF PAGE HEADERS NOT DESIRED ----
           write(42,960)s(j,13),s(j,14),s(j,15)
960        format("+",t117,a4,"-",i2,"-",a2)
970        if(.not.lnm) go to 1010
           if(ilt(j).eq.0) go to 1010
c
c ---- EXPANDS INTEGER LAT/LON TO DEGREE AND DECIMAL MINUTE VALUES ----
           rlt=float(ilt(j))/100000
           xlt=aint(rlt)
           xltm=(rlt-xlt)*60
           lat=ifix(xlt)
           rln=float(iln(j))/100000
           xln=aint(rln)
           xlnm=abs((rln-xln)*60)
           lon=ifix(xln)
           if(.not.lhs) go to 990
c  ---- WRITES OUT LAT/LON FOR TIMESORT ----
           write(42,980) lat,xltm,lon,xlnm
980        format("+",t74,i3,1x,f5.2,1x,i4,1x,f5.2)
           go to 1010
c ---- WRITES OUT LAT/LON FOR HIERSORT ----
990        write(42,1000)lat,xltm,lon,xlnm
1000       format("+",t95,i3,1x,f5.2,1x,i4,1x,f5.2)
c ---- NRO COUNTS TOTAL RECORD OUTPUT ----
1010       nro=nro+1
```

```
c ---- NLO COUNTS LINES OF OUTPUT ON A GIVEN PAGE ----
1020      nlo=nlo+1
          return
          end
```

```
COMPILATION LISTING OF SEGMENT finpun
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1637.5 pst Tue
    Options: table symbols

          subroutine finpun
          logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm
          ,lout
          real idt
          double precision sv,svk
          character*4 isn,rb
          character*8 cs1,cs2,date,dout,dsh1,dsh2,ibk,ids,ist,mout,new,newl,
          shp1,shp2
          integer cod,dy,hi,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nsl,s,sp
          common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
          s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
          ,im(2400)
          common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
          nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
          ,lie,lpout,lnm,lout
          common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
          shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
          med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
          l4(158)
c
          il=0
          jump=1
c ---- LPOUT = T => OUTPUT IS FOR OUTSIDE USE (NGSDC) ----
          if(.not.lpout) jump=2
c
c
c
c ---- LOOP PROCESSESING AND CONDENSING CDI RECORD VARIABLES TO AN 80 COLUMN
c        CARD FORMAT ----
          do 112 i=1,n
          j=i
          if(lsi) j=ni(i)
          if(lnm) go to 4
          ilat=0
          ilon=0
          go to 8
4         ilat=ilt(j)
          ilon=iln(j)
8         jm=im(j)
          nn=3
c
c ---- CONVERTING ST/ED/ON/OFF INDICATOR TO THE NUMBERS 1/2/3 OR 4 ----
          do 12 m=1,4
          nn=nn+1
          if(s(j,nn).ne.ib) go to 16
12        continue
          ista=0
          go to 20
16        ista=m
c ---- TRANSLATES POSITION OF MEDIA CODE IN DATA STATEMENT MED, TO ANOTHER
c        NUMBER SEQUENCE ----
20        mh=hm(jm)
          go to (40,22),jump
c
c
```

```
c ---- SECTION TRANSLATING CDI RECORD TO INHOUSE STORAGE DECK OR FILE FORMAT ----
c ---- TRANSLATES NUMBER DENOTING POSITION OF DATA CODE IN DATA STATEMENT
c        'COD', TO A HEXIDECIMAL NUMBER ----
22         icx2=ic(j)/16
           icx1=ic(j)-icx2*16
           if(icx1.eq.0) icx1=18
           icx1=idas(icx1)
c ---- TRANSLATES POSITION OF MEDIA CODE IN MED TO HEXIDECIMAL NUMBER --
           imx=idas(jm)
           if(jm.eq.16) imx=idas(1)
           imn=rmn(j)*10
c ---- WRITE OUT CONDENSED RECORD TO FILE12 IN WORKING DIRECTORY ----
           write(12,32)dy(j),hr(j),imn,lin(j),sp(j),imx,icx2,icx1,s(j,3),isn(
           j),ista,(idd(j,l),l=1,6),(s(j,l),l=8,12),iu(j),ilat,ilon,(s(j,l),
           l=13,15)
32         format(i3,i2,i3,2a4,a1,i1,2a1,a4,i1,6a4,a1,4i1,a1,i8,i9,a4,i2,a2)
           go to 112
c
c
c ---- SECTION TRANSLATING CDI RECORDS TO NGSDC FORMAT ----
40         if(mh.lt.11) go to 48
c ---- WRITES OUT ALL CRUZ, PORT, PER AND EQP CDI RECORDS ----
           write(12,44)dy(j),hr(j),rmn(j),(s(j,l),l=16,19),ista,isn(j),s(j,2)
           ,(idd(j,l),l=1,6),(s(j,l),l=13,15)
44         format(i3,i2,f4.1,t18,4a1,i1,1x,a3,a4,6a4,t73,a4,i2,a2)
           go to 112
48         hi=s(j,20)
c ---- GEOLOGICAL AND HYDROLOGICAL RECORDS TO 62 ----
           if(hi.gt.299999) go to 62
c ---- STATION RECORDS TO 82 ----
           if(hi.eq.108110) go to 82
c
c ---- PROCESSES GEOPHYSICAL AND NAVIGATION RECORDS ----
           jn=jm-3
           if(jn.eq.-2) jn=13
           iq=ista
           if(iq.gt.2) go to 52
           go to 56
52         iq=9
           if(ista.eq.4) iq=10
c ---- WRITES OUT GEOPHYSICAL AND NAVIGATION RECORDS ----
56         write(12,58)dy(j),hr(j),rmn(j),lin(j),sp(j),(s(j,l),l=16,19),ista,
           s(j,3),isn(j),s(j,2),new(jn),s(j,3),isn(j),ist(iq),ilat,ilon,(s(j,
           l),l=13,15)
58         format(i3,i2,f4.1,2a4,4a1,i1,a1,2a4,a8,a1,a4,t46,a8,2x,i8,i9,a4,i2
           ,a2)
           go to 112
c
c ---- PROCESSES ALL GEOLOGIC AND HYDROGRAPHIC RECORDS ----
62         if(mh.ne.8) go to 86
c
c ---- PROCESSES SAMPLES ----
           if(ista.eq.0) go to 70
           jn=18
           if(ista.eq.2) go to 66
c ---- PLACES 'START' IN OUTPUT VARIABLE IDS (FOR DREDGES) ----
           ids=ist(11)
           go to 94
c ---- PLACES 'END R' IN OUTPUT VARIABLE IDS (FOR DREDGES) ----
66         ids=ist(14)
```

```
c ---- IF NO RECOVERY INDICATED, PLACES 'END NO R' IN IDS (DREDGES) ----
           if(s(j,8).ne.ir) ids=ist(15)
           go to 94
c ---- PLACES 'RECOV' IN IDS ----
70         ids=ist(12)
c ---- IF NO RECOVERY INDICATED, PLACES 'NRECO' IN IDS ----
           if(s(j,8).ne.ir) ids=ist(13)
           if(hi.gt.399999) go to 74
c ---- GEOLOGIC RECORDS ----
           jn=hi/1000-300+16
           if(jn.eq.23) jn=24
           go to 94
c ---- HYDROGRAPHIC RECORDS ----
74         ihi=hi/100-4000
           if(ihi.gt.2) go to 78
           jn=21
           go to 94
78         jn=22
           if(ihi.eq.50) jn=23
           go to 94
82         jn=14
           go to 90
c
c ---- PROCESSES NON-SAMPLES ----
86         jn=jm-3
           if(ista.eq.0) ids=ibk
90         ids=ist(ista)
           if(ista.eq.1) ids=ist(11)
c
c ---- WRITES OUT REMAINING RECORDS ----
94         write(12,96) dy(j),hr(j),rmn(j),lin(j),sp(j),(s(j,l),l=16,19),ista
           ,s(j,3),isn(j),s(j,2),new(jn),s(j,3),isn(j),ids,(s(j,l),l=9,12),iu
           (j),ilat,ilon,(s(j,l),l=13,15)
96         format(i3,i2,f4.1,2a4,4a1,i1,a1,2a4,a8,a1,a4,t46,a5,i4,a1,i8,i9,a4
           ,i2,a2)
112        continue
116        return
           end
```

```
COMPILATION LISTING OF SEGMENT title
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1637.8 pst Tue
    Options: table symbols

          subroutine title (shp1,shp2,cs1,cs2,date,s,lhs,lnm,nz,j)
          integer s
          logical lhs,lnm
          character*8 shp1,shp2,cs1,cs2,date
          dimension shp1(5),shp2(5),cs1(5),cs2(5),date(2),s(2400,20)
c
          write(42,1)
1         format(//////////////)
          write(42,4)
4         format(" ",t40,"HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH%
HHHH")
          write(42,7)
7         format(" ",t40,"H",t93,"H",/t40,"H",t93,"H")
          write(42,10)
10        format(" ",t40,"H",t58,"CRUISE DATA INDEX",t93,"H")
          write(42,13)
13        format("+",t58,"______ ____ _____")
          write(42,7)
          write(42,19)
19        format(" ",t40,"H",t65,"FOR",t93,"H")
          write(42,22)
22        format(" ",t40,"H",t93,"H")
          write(42,25) (s(j,l),l=13,15)
25        format(" ",t40,"H",t60,"- ",a4,"-",i2,"-",a2," -",t93,"H")
          write(42,7)
          write(42,27)
27        format(" ",t40,"H",t66,"ON",t93,"H"/t40,"H",t63,"THE SHIP",t93,"H"
)
          write(42,29) shp1(nz),shp2(nz)
29        format(" ",t40,"H",t59,2a8,t93,"H")
          write(42,22)
          write(42,29) cs1(nz),cs2(nz)
          write(42,31)
31        format("+",t58,"__________________")
          write(42,34)
34        format(" ",t40,"H",t58,"CHIEF SCIENTIST(S)",t93,"H")
          write(42,7)
          if(.not.lnm) go to 40
          write(42,37)
37        format(" ",t40,"H    (INCLUDES NAVIGATION FIXES FOR DATA ENTRIES) %
   H")
          go to 46
40        write(42,43)
43        format(" ",t40,"H     (NAVIGATION FIXES FOR DATA ENTRIES PENDING) %
   H")
46        write(42,22)
          write(42,49)
49        format(" ",t40,"H",t50,"LISTING GENERATED: ______________",t93,"H"
)
          if(.not.lhs) go to 55
          write(42,52)
52        format("+",t70,"BY DATA TYPE")
          go to 61
55        write(42,58)
58        format("+",t69,"SORTED BY TIME")
```

```
61         write(42,22)
           write(42,64) date
64         format(" ",t40,"H",t56,"WHEN: ",2a8,t93,"H")
           write(42,67)
67         format("+",t61,"__________________")
           write(42,22)
           write(42,4)
           write(42,70)
70         format("1")
           write(42,70)
           return
           end
```

```
COMPILATION LISTING OF SEGMENT block_data
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1638.3 pst Tue
    Options: table symbols

          block data
          logical ldt,labnd1,lav,lts,lsi,lci,lhs,lfo,lph,lfp,lie,lpout,lnm,lout
          real iot
          double precision sv,svk
          character*4 cmnd,isn,rb
          character*8 cs1,cs2,date,dot1(119),dot2(119),dot3(78),dout,dsh1,
          dsh2,ibk,ids,ist,mout,new,newl,shp1,shp2
          integer cod,dy,hi,hir,hm,hr,ic,im,ios,iu,l1,l2,l3,l4,n,ni,nsl,s,sp
c
c ---- LABELED COMMON STATEMENTS CONTAINING VARIABLES USED BY MOST
c         SUBROUTINES OF EDIT CDI ----
c ---- INP CONTAINS INPUT CDI RECORD VARIABLES ----
          common/inp/rmn(2400),lin(2400),sp(2400),isn(2400),idd(2400,6),
          s(2400,20),ilt(2400),iln(2400),dy(2400),hr(2400),iu(2400),ic(2400)
          ,im(2400)
c ---- DTA CONTAINS LOGICALS AND DATA STATUS INDICATORS ----
          common/dta/sv(2400),dsh2,ibk,rb,ib,idt,is,ir,in,ix,nro,ncpds,
          nss,idas(18),ni(2400),n,lav,labnd1,ldt,lts,lci,lhs,lph,lfp,lfo,lsi
          ,lie,lpout,lnm,lout
c ---- SER CONTAINS MEDIA AND DATA CODES, THEIR ENGLISH EQUIVALENTS, HIERAR-
c         CHAL NUMBERS, LADDER CODES, ETC. ----
          common/ser/new(24),dout(316),mout(30),ist(17),dsh1(4),shp1(5),
          shp2(5),cs1(5),cs2(5),date(2),ks(100),ida(5),cod(158),hi(158),
          med(15),cmnd(9),nsl(10),ios(100),hm(17),l1(5),l2(17),l3(158),
          l4(158)
          equivalence (dout(1),dot1(1)),(dout(120),dot2(1)),(dout(239),dot3(
          1))
c
c ---- DATA STATEMENT CONTAINING TRANSLATIONAL NUMBERS CORRESPONDING TO
c         POSITION OF MEDIA CODES IN MED ----
          data hm/10,12,13,8,1,2,0,0,3,9,4,5,0,6,7,11,0/
c
c ---- DATA STATEMENT CONTAINING MEDIA CODES ----
          data med /"NAV","PER","EQP","SAM","PAP","MAG","SYS","OBS","AVM",
          "PHO","PRN","DEP","MAP","CRD","PUN"/
c
c ---- DATA STATEMENT CONTAINING 1ST LEVEL LADDER CODE CORRESPONDING TO
c         GEOPHYSICAL, GEOLOGIC, VESSEL OR HYDROGRAPHIC DATA ----
          data l1 /"X","G","V","H"," "/
c
c ---- DATA STATEMENT CONTAINING 2ND LEVEL LADDER CODES.  DENOTES MEDIA TYPE
c         OR A MORE GENERAL DATA CATEGORY THAN 3RD LEVEL LADDER CODE ----
          data l2 /"K","R","W","S","A","M","X","O","T","P","G","D","H","C",
          "F","K"," "/
c
c ---- DATA STATEMENT CONTAINING 3RD LEVEL LADDER CODES.  DENOTES A GENERAL
c         DATA CATEGORY ----
          data l3 /"N","N","Y","Y","Y","R","R","R","R","R","R","R","R","R",
          "R","R","R"," "," ","N","N","N","N","N","N","N","N","N","N","N",
          "N","N"," "," ","N","N","N","N","N"," "," "," ","T","T","X","X",
          "X","X"," "," ","T","T","T","T"," ","Q","Q","Q","W","W","W","H",
          "H","H","H","H"," ","B","B","B","B","B","B","B","B","M","Z","M",
          "M","M","G","G","G","G","G","G","J","K","F","C","C","C","C","C",
          "C","C","C","C","C","C"," ","K","K","K","K","K","K","K","K",
          "K","K"," ","D","D","U","O","Q","Q"," ","J","L","E","M","W","W",
          "W","W"," "," "," ","T","T","T"," "," ","C","C","C","C","C"," ",
          " ","L"," "," ","P","K","K","N","K","S"," ","I","I","I"," "," ",
          " "/
c
c ---- DATA STATEMENT CONTAINING 4TH LEVEL LADDER CODES.  DENOTES A SPECIFIC
c          DATA TYPE ----
          data l4 /"H","P","C","E","M","S","D","G","X","Y","B","O","Q",
          "T","W","N","U"," "," ","S","D","R","M","V","6","A","C","Q","X",
          "Y","F","L"," "," ","Z","B","W","T","K"," "," "," ","A","X","R",
          "K","H","P"," "," ","M","A","X","5"," ","A","X","5","O","F","E",
          "U","Z","D","S","7"," ","I","X","Y","W","N","T","8","9","M","Z",
          "4","V","L","G","2","1","3","6","B","T","N","C","D","G","B",
          "P","R","T","A","S","N","F","L"," ","V","K","F","X","Y","Z","O",
          "M","U","E"," ","C","W"," "," ","I","J"," ","Q"," ","H","M","N",
          "K","V","W"," "," "," ","T","P","I"," "," ","R","D","X","M","H",
          " "," ","A"," "," ","S","E","Q","L","Z","F"," ","B","C","G"," ",
          " "," "/
c
c ---- DATA STATEMENT CONTAINING DATA CODES FOR EACH DATA TYPE. POSITION OF
c          CODE IN THE ARRAY SIGNALLED BY IC OR JC ----
          data cod /"CRUZ","PORT","CAPT","CENG","MATE","KING","DAFE","GEOL",
          "GEOP","GEOC","BIOL","OCEO","ETEC","MTEC","WACH","NAVG","UNSP",
          "    ",
          "    ","SATN","DOPP","RADR","MRAN","NORT","SINS","LORA","LORC",
          "LORR","HFIX","RDIS","CREC","LORK","    ","    ","DRSL","DRBR",
          "DRCP","TRAK","STAT","    ","    ","    ","AGUN","SPRK","MC24",
          "MC36","MC48","MC72","    ","    ","SCAM","SCAG","SCAR","SCAA",
          "    ","BRGA","BRGS","BRG2","BOTS","SONO","BOOM","UNIB","MSPK",
          "DELN","SIDE","UBMS","    ","DIGT","BA35","BA7K","BA12",
          "BA48","BA2H","BATS","BATN","MAGS","GRAD","MAGB","MAGN","MABN",
          "GRVS","GRVB","GRVM","GRV2","GRV3","GRBN","GRVL","MAGL","HEAT",
          "DART","GRAV","BOXS","PSTN","VIBR","DRIL","ALPN","RANG","BOXH",
          "FLAG","HYPC","    ","VANV","SHIP","FREE","PETR","PONR","CAMP",
          "PEEL","HAND","SUTR","DETZ","    ","CHAN","PIPE","UWAY","SUSP",
          "TVIS","CMRA","    ","SDIV","LAND","PENT","MARK","NISK","NANS",
          "VAND","WATB","    ","    ","    ","TRAN","NEPH","SECC","    ",
          "    ","ANDR","GEOD","BEND","MARS","HYDP","    ","    ","ATYD",
          "    ","    ","WAVE","XBTG","TSAL","PLAN","TEMP","RIPL","    ",
          "BECK","CTDM","GEOP","    ","    ","    "/
c
c ---- DATA STATEMENTS CONTAINING HIERARCHAL CODES FOR EACH DATA TYPE.  IN
c          SAME ORDER AS DATA CODES IN COD.  POSITION IN ARRAY INDICATED BY IC
c          OR JC ----
          data hi /100000,100000,101000,101100,101200,101300,101400,
          101500,101600,101700,101800,101900,102000,102100,102200,102300,
          102400,      ,      ,105000,105100,105200,105300,105400,105500,
          105600,105700,105800,105900,106000,106100,106200,000000,000000,
          106500,
          106600,106700,108000,108100,      ,000000,000000,200000,200100,
          201000,201100,201200,201300,000000,000000,202000,202100,202200,
          202300,000000,203000,203100,203200,204000,204100,204200,205000,
          205100,205200,205300,205400,000000,206000,206100,206200,206300,
          206400,206500,206600,206700,207000,208000,208100,208200,208300,
          210000,210100,210200,210300,210400,210500,211000,209000,220000,
          300000,300100,300200,300300,300400,300500,300600,300700,300800,
          300900,300910,000000,301000,301100,301200,301300,301400,301500,
          301600,301700,301800,301900,000000,302000,302100,303000,304000,
          305000,305100,000000,306000,307000,308000,309000,400000,400100,
          400200,400300,000000,000000,000000,401000,401100,401200,000000,
          000000,402000,402100,402200,402300,402400,000000,000000,403000,
           000000,000000,404000,404100,404200,405000,404300,406000,000000,
           407000,407100,407200,000000,000000,000000/
c
c ---- COMMANDS USED IN CDIMAIN TO INVOKE VARIOUS FUNCTIONS AND SUBROUTINES
c         OF EDIT CDI ----
           data cmnd /"inpu","auto","sort","sear","merg","prin","punc","outp"
           ,"stop"/
c
c ---- DATA STATEMENTS CONTAINING LITERALS FOR ST/END/ON/OFF INDICATORS,
c         RECOVERY/NO RECOVERY INDICATORS, ETC. ----
           data is,ir,in,ix,ib,ibk,idt/"YES","Y","N","X","    ","        ",
           "UGS"/,ist/"START","END","ON","OFF","RECOVERY"," NO RECO",
           "END RECO","END NO R","DATA ON ","DATA OFF","STRT ","RECOV",
           "NRECO","END R","END N"," ON "," OFF"/,dsh1,dsh2/5*"--------"/,
           idas/"1","2","3","4","5","6","7","8","9","A","B","C","D","E","F",
           "V"," ","0"/
c
c ---- DATA STATEMENT CONTAINING MEDIA OR GENERAL DATA TYPE DESCRIPTIONS
c         CORRESPONDING TO MEDIA CODES AND LINKED BY JN ----
           data new/"  SAMPLE","    ROLL","    REEL","  SYSTEM","READINGS",
           "    REEL","   PHOTO","    LIST","  NUMBER","MAP/PLOT","CRD DECK",
           "    ROLL","    LINE"," STATION","        ","    CORE","    GRAB",
           "  DREDGE"," SAMPLER","SEDIMENT","  BOTTLE","  BUCKET","     NET",
           "    LAND"/
c
c ---- DATA STATEMENT CONTAINING MEDIA OR GENERAL DATA GROUP DESCRIPTORS ----
           data mout/"        ","        ","PERSONNE","L LIST","EQUIPMEN",
           "T LIST","SAMPLE A","TTEMPT","ANL PAPE","R ROLL","DIGIT MA",
           "G TAPE","OPRTING ","PERIOD","NUM.OBSR","VATION","ANLOG MA",
           "G TAPE","    PHOT","OGRAPH","PRINTR L","ISTING","    DEPL",
           "OYMENT","   MAP O","R PLOT","COMP CAR","D DECK","DIG PUNC",
           "H TAPE"/
c
c ---- DATA STATEMENT DOUT (EQUIVALENT TO DOT1, DOT2, AND DOT3) WHICH CONTAINS
c         DATA TYPE DESCRIPTION CORRESONDING TO EACH DATA CODE FOUND IN COD ----
           data dot1/"CRUISE  ","        ","IN PORT ","        ","SHIP CAP",
           "TAIN    ","CHIEF EN","GINEER  ","CHIEF MA","TE      ","CHIEF SC",
           "IENTST  ","DAFE CUR","ATOR    ","GEOLOGIS","T       ","GEOPHYSI",
           "CIST    ","GEOCHEMI","ST      ","BIOLOGIS","T       ","OCEANOGR",
           "APHER   ","ELECTRON","ICS T   ","MECHANIC","AL T    ","WATCH ST",
           "ANDER   ","NAVIGATO","R       ","UNSP INV","ESTIGATR","        ",
           "        ","        ","        ","NAV SATE","LLITE   ","DOPPLER ",
           "SONAR   ","RADAR   ","        ","MINIRANG","ER      ","DEL NORT",
           " RNG/RNG","INTEGRAT","ED NAV  ","LORAN A ","        ","LORAN C ",
           "        ","LORAN C ","RHO-RHO ","HI-FIX  ","        ","RAY DIST",
           "        ","COURSE R","ECORDER ","LORAC NA","V       ","        ",
           "        ","        ","        ","DR NAV P","LOT/LAB ","DR NAV P",
           "LOT/BURG","DR NAV P","LOT/COMP","     TRA","CKLINE  ","       S",
           "TATION  ","        ","        ","        ","        ","        ",
           "        ","AIRGUN  ","        ","SPARKER ","        ","24 CHANL",
           " SEISMIC","36 CHANL"," SEISMIC","48 CHANL"," SEISMIC","72 CHANL",
           " SEISMIC","        ","        ","        ","        ","MULT CHA",
           "N MONITR","SNGL CHA","N AIRGUN","SNGL CHA","NL ARCER","S CHA AI",
           "RGN/ARCR","        ","        ","BURG AIR","GUN     ","BURG SPA",
           "RKER    ","BRIG AIR","GN/SPRKR","BOTTOM S","EISMOMTR","SONOBUOY"/
           data dot2/"        ","EXPLOSIV","ES      ","UNIBOOM ","        ",
           "MINISPAR","KER     ","DEL NORT","E       ","SIDE SCA","N SONAR ",
           "UNIBOOM/","MINSPRKR","        ","        ","DIGITRAC","K       ",
           "3.5KH BA","THYMETRY","7 KHZ BA","THYMETRY","12KH BAT","HYMETRY ",
           "48KH BAT","HYMETRY ","200KH BA","THYMETRY","200 + 7K","H BATHYM",
      "BATHYMET","RY + NAV","SHIPBOAR","D MAGGY ","GRADIOME","TER     ",
      "MAGGY + ","BATHYMET","MAGNETIC","S + NAV ","MAG/BATH","YM/NAVIG",
      "SHIPBOAR","D GRAVTY","GRAVITY ","+ BATHYM","GRAVITY ","+ MAGNET",
      "GRAV/MAG","/BATHYM ","GRA/MAG/","BATH/NAV","GRAV/BAT","HY NAVIG",
      "STA GRAV"," LAND TY","LAND STA","TION MAG","HEAT FLO","W GAUGE ",
      "DART COR","E       ","GRAVITY ","CORE    ","BOX(SHIP",") CORE  ",
      "PISTON C","ORE     ","VIBRATIN","G CORE  ","DRILL CO","RE      ",
      "ALPINE C","ORE     ","BOOMERAN","G CORE  ","HAND BOX"," CORE   ",
      "PHLEGER ","CORE    ","HYDROPLA","ST. CORE","        ","        ",
      "VAN VEEN"," GRAB   ","SHIPECK ","GRAB    ","FREEFALL"," GRAB   ",
      "PETERSON"," GRAB   ","PONAR GR","AB      ","CAMPBELL"," GRAB   ",
      "ORANGE P","EEL GRAB","HAND SAM","PLE GRAB","SOUTAR G","RAB     ",
      "DIETZ/LF","OND GRAB","        ","        ","CHAIN DR","EDGE    ",
      "PIPE DRE","DGE     ","UNDERWAY"," SAMPLER","SUSPEND ","SEDIMENT",
      "TELEVISI","ON      ","SEAFLOOR"," CAMERA ","        ","        "/
      data dot3/"SCUBA DI","VE      ","LAND SAM","PLE     ","PENETROM",
      "ETER    ","SIDE SCA","N MARKER","NISKEN B","OTTLE   ","NANSEN B",
      "OTTLE   ","VAN DORN"," BOTTLE ","WATER BU","CKET    ","        ",
      "        ","        ","        ","        ","        ","TRANSMIS",
      "SOMETER ","NEPHELOM","ETER    ","SECCHI D","ISK     ","        ",
      "        ","        ","        ","AND CURR","ENT METR","GEO CURR",
      "ENT METR","BEN CURR","ENT METR","MAR CURR","ENT METR","HYD CURR",
      "ENT METR","        ","        ","        ","        ","ANDERA T",
      "IDE METR","        ","        ","        ","        ","WAVE PRE",
      "S SENSOR","EXP BATH","Y THERMO","TEMP/SAL","INOMETER","PLANKTON",
      " NET    ","TEMPRATU","RE GAUGE","RIPPLE P","ROFILER ","        ",
      "        ","BECKMAN ","STC     ","CTD METE","R       ","GEOPROBE",
      "        ","        ","        ","        ","        ","        ",
      "        "/
      end
```

```
PROGRAM STATCDI



PROGRAM STATCDI - STATISTICS ON CRUISE DATA INDEX
  APPLICATION   - SPECIFIC TO CRUISE DATA INDEX
  FUNCTION      - COMPILES STATISTICS, IE., TRACKLINE DISTANCES, DATA
                     QUANTITIES, ETC., FROM CRUISE DATA INDICES
  MODE          - INTERACTIVE


=========================================================================
                      I D E N T I F I C A T I O N
=========================================================================

  PROGRAM.......STATCDI

  DESCRIPTION...THIS PROGRAM HELPS TO EDIT A CRUISE DATA INDEX AS WELL
                AS CONDENSE IT INTO A CRUISE REPORT FORMAT WHICH LISTS
                CRUISE INFORMATION, DATA QUANTITIES, AND DATA SYSTEM
                TRACKLINE DISTANCES AND RECORDING HOURS.  SPECIFIC
                FUNCTIONS OF STATCDI INCLUDE:
                (1) FLAGS ANY DISRUPTION OF PROPER 'START/END/ON/OFF'
                    SEQUENCES FOR EACH DATA TYPE.
                (2) LISTS BROAD DATA CATEGORIES COLLECTED ON A CRUISE,
                    WHETHER GEOLOGIC, GEOPHYSICAL, ETC.
                (3) CONVERTS JULIAN DAY AND GMT FOR 'START/END' OF
                    CRUISE AND PORT STOPS TO LOCAL STANDARD DATES AND
                    TIMES.
                (4) CALCULATES TOTAL UNDERWAY AND TOTAL PORT TIMES.
                (5) LISTS PERSONNEL, THEIR AFFILIATIONS AND DUTIES,
                    AND WHEN THEY CAME ABOARD OR LEFT.
                (6) LISTS ALL DATA EQUIPMENT SYSTEMS USED, BY
                    CATEGORY.
                (7) LISTS ALL GEOPHYSICAL DATA COLLECTED.  THIS
                    INCLUDES THE RECORDING MEDIUM, THE QUANTITY OF
                    PAPER ROLLS, TAPES, ETC., THE RECORDING HOURS, AND
                    THE TRACKLINE KILOMETERS AND NAUTICAL MILES FOR
                    EACH SYSTEM.
                (8) LISTS ALL GEOLOGIC/HYDROLOGIC SAMPLES COLLECTED.
                    THIS INCLUDES THE NUMBER OF SAMPLE ATTEMPTS, THE
                    NUMBER OF SAMPLES RECOVERED, THE NUMBER RECOVERED
                    FROM SPECIFIC DEPTH INTERVALS AND THE TOTAL NUMBER
                    IN EACH OF THE FOREGOING CATEGORIES.
                (9) LISTS ALL ANALOG GEOLOGIC/HYDROLOGIC DATA
                    COLLECTED ALONG WITH THEIR RECORDING TIMES AND
                    QUANTITIES.
                (10) LISTS THE NUMBER AND TYPE OF NUMERICAL
                    OBSERVATIONS TAKEN AND OVER HOW MANY STATIONS.
                (11) LISTS OPERATIONS INFORMATION INCLUDING NUMBER OF
                    STATIONS OCCUPIED, TOTAL TIME ON STATION, NUMBER
                    OF TRACKLINES RUN, TOTAL TRACKLINE TIME,
                    CUMULATIVE TRACKLINE DISTANCES, AND THE NUMBER OF
```

```
PROGRAM STATCDI

                    DEPLOYMENTS OF SONOBUOYS, GEOPROBES, ETC.

  VERSION.......28 SEP 1977

  CONTACT.......TOM CHASE, BRADLEY LARSEN
                MARINE GEOLOGY, U.S. GEOLOGICAL SURVEY, MENLO PARK

  PROGRAMMING...B. LARSEN  ORIGINAL                         MAR 1977
                    STAT, DIST
                B. LARSEN  MODIFICATION                     APR 1977
                    ADD PRINT, SORT, DATCON, BLOCKDATA

  COMPUTERS.....HONEYWELL SERIES 60 UNDER MULTICS
  PROCESSORS....HONEYWELL MULTICS FORTRAN (RELEASE 4)

  INPUT.........(1) ANSWERS TO INTERACTIVE PROMPTS (TERMINAL)
                (2) EXPANDED CDI LOG ENTRIES WITH DECIMAL MINUTES AND
                    'ADDITIONAL VARIABLES'; OR DECIMAL MINUTES,
                    'ADDITIONAL VARIABLES' AND NAVIGATION (UNIT 5)
                (3) NAVIGATION DATA FILES IN FORMATTED CARD IMAGE
                    (UNIT 10), OR UNFORMATTED FROM MAGNAVOX TAPE (UNIT
                    10)

  OUTPUT........(1) CRUISE REPORT FORMATS FOR WHEN NAVIGATION FILES
                    ARE USED AND WHEN THEY ARE NOT USED.

  SUBROUTINES...(A) STAT      GROUPS DATA, CALCULATES DISTANCES,
                              QUANTITIES AND RECORDING TIMES FOR EACH
                              GROUP
                (B) DIST      CALLED BY STAT. CALCULATES TRACKLINE
                              DISTANCES BASED ON TRIG FUNCTIONS OR
                              RECORDING TIMES AND AVERAGE SHIP SPEED.
                (C) PRINT     PRINTS INFORMATION DETERMINED IN STAT AND
                              DIST IN CRUISE REPORT FORMAT.
                (D) DATCON    CONVERTS JULIAN DAY AND GMT TO LOCAL
                              STANDARD DATES AND TIME.
                (E) SORT      HIERARCHALLY SORTS EACH GROUP OF DATA.
                (F) BLOCKDATA STORES LABELED COMMON AND DATA
                              STATEMENTS USED IN PRINT.

  EXECUTION.....PROGRAM CAN BE RUN BY INVOKING: rprt$stat
```

PROGRAM STATCDI

OPERATION

EXAMPLE TYPE

The following example is given to show how to initiate and direct a STATCDI program run on a CDI input data set which has had navigation positions merged with it. The CDI data set is virtually error-free except for the first 3.5KH Bathymetry record which is erroneously marked as an 'end' record instead of a 'start' record. The input navigation data set is assumed to be on File 3 of ansi tape S6NCL2.

PRECAUTIONS

(1) The user should be aware of the following information before initiating a program run:
  (A) Did the CDI use NAV/STAT record entries to mark the start and end of stations?
  (B) If sample records are present, were station numbers placed in the 'sp' or shot point/station number field?
  (C) What is a more descriptive location of the area of the cruise which generated the input CDI?
  (D) What are the names of the Chief Scientists?
  (E) What are the longitudinal meridians of all the ports the ship was at during any part of the cruise?

(2) Navigation data sets on tape generated by the Magnavox SATNAV system must be set up using 'prefio' before STATCDI is called (example below). Otherwise, 'prefio' may be skipped.

(3) In the example, interactive prompts or messages are prefaced by a dash (-); system responses are prefaced by a bracket ([).

PROGRAM RUN

TAPE_SET-UP

```
   prefio
    -Enter switch number, eg, 09
   10
    -Enter name for dummy segment (max. 6 chars):
   dum
    -Input or output action (i/o)?
   i
```

PROGRAM STATCDI

```
         [r 1605 0.552 7.780 126
        io attach file10 tape_ansi_ S6NCL2 -nm F3
         [Mounting volume S6NCL2 with no write ring.
         [S6NCL2 mounted on tape_03
         [r 1607 1.180 12.224 174
        io open file10 sqi
         [r 1607 0.347 2.178 56

     INITIATE_STATCDI
        rprt$stat
         -WERE STATIONS SPECIFICALLY ENTERED IN sp OR AS CDI
         ENTRIES?? (t or f)(l1)
        t
         -??WERE STATIONS ENTERED IN CDI AS -NAV STAT- ?? (t OR
         f)(l1)
        t
         -ENTER INPUT SEGMENT NAME - UP TO 32 CHARACTERS.
        sea376tsnm
         -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
        Cruise_Report.sea376
         -??WILL THIS PROGRAM-RUN USE NAVIGATION FILES?? (t OR f)(l1)
        t
         -??HAS NAVIGATION BEEN ADDED TO CDI?? (t OR f)(l1)
        t
         -IF NAV HAS NOT BEEN MERGED W/CDI, PROGRAM TERMINATES.
         -??IS MAG. INPUT ON MAGNAVOX TAPE?? (t OR f)(l1)
        t

     WARNING_MESSAGES
         -FIRST    RECORD OF DATA TYPE -PAP BA35- IS NOT A START/ON.
          TIME/DX ACCUMULATION WILL NOT BEGIN UNTIL A START/ON RECORD
          IS ENCOUNTERED.  DAY/TIME FOR THE FIRST   RECORD IS: 189/
          218.0
         -SECOND   RECORD OF DATA TYPE -PAP BA35- IS NOT A START/ON.
          TIME/DX ACCUMULATION WILL NOT BEGIN UNTIL A START/ON RECORD
          IS ENCOUNTERED.  DAY/TIME FOR THE SECOND   RECORD IS: 189/
          826.0
         -BAD NAV FOR RECORD PAP BA35 OF DAY/TIME 196 020.0 TREATED
          AS IF NAV=0
         -BAD NAV FOR RECORD SAM DART OF DAY/TIME 205/1027.0 TREATED
          AS IF NAV=0
         -BAD NAV FOR RECORD SAM DART OF DAY/TIME 205/2220.0 TREATED
          AS IF NAV=0

     PROMPTS_FROM_'PRINT'_SUBROUTINE
         -GIVE A MORE EXACT LOCATION OF THIS CRUISE.  USE 40
          CHARACTERS OR LESS.  E.G., POINT MENDECINO CAL. TO COOS BAY
          ORE.; MONTEREY-DELGADO FAN; BEAUFORT SEA; COOK INLET TO
          KODIAK ISLAND;  NORTON SOUND, NORTHERN BERING; ETC.
        COOK INLET TO KODIAK ISLAND
         -ENTER THE NAME(S) OF THE CHIEF SCIENTIST(S) FOR THIS
          CRUISE.  IF 2 INDIVIDUALS INVOLVED, SEPARATE THEIR NAMES
```

```
  WITH A /. (4a8)
A. BOUMA / M. HAMPTON
 -ENTER THE LOGITUDINAL MERIDIAN OF THE PORTS WHERE SHIP
  BEGINS CRUISE, WHERE IT STOPS MID-CRUISE, AND WHERE IT ENDS
  CRUISE.  USE DECIMAL DEGREES W/OUT (-) OR (+) PREFIX. E.G.,
  163.12345
 -ENTER MERIDIAN FOR PORT:  LV HOMER AK, START CRUIS ,AFTER>>
 ->>
151.30000
 -ENTER MERIDIAN FOR PORT:  ARR KODIAK AK
 ->>
152.45000
 -ENTER MERIDIAN FOR PORT:  KOKIAK AK ED CRUISE
 ->>
152.45000
 -LIST OF NAMES LOGGED ONLY ONCE FOLLOWS, CHECK SPELLING
  INCONSISTENCIES:
    1 MCCLENAGHAN, A.             JD   0/  0
    2 SHEPPARD, H.                JD   0/  0
    3 JOHANNESSEN, O.             JD   0/  0
    4 GARDENR, JAMES              JD 218/  0
    5 VALLIER, TRACY              JD 218/  0
    6 GARDNER, JAMES              JD   0/253
    7 VALLUR, T.                  JD   0/253
    8 KLISE, DAVID                JD 218/  0
    9 KLEISE, DAVE                JD   0/253
 -??HOW MANY PAIRS OF DUPLICATE NAMES ARE THERE?? (i2
  W/LEADING ZEROS)
03
 -FOR EACH PAIR OF DUPLICATE NAMES, ENTER THEIR NUMBERS FROM
  THE ABOVE LIST.  ENTER THEM TWO AT A TIME LISTING THE
  NUMBER TO BE DELETED, FIRST. (i2,1x,i2) FORMAT W/LEADING
  ZEROS. ENTERING 03 04 MEANS NAME 3 WILL BE DELETED.
  ENTRIES PROMPTED BY >>
 ->>
04 06
 ->>
07 05
 ->>
09 08
 [STOP
 [fortran_io_: Close files?  yes
 [r 1710 61.567 39.174 591
```

RESULTS

This program run will produce one data set segment in the user's working directory: Cruise_Report.sea376

PROGRAM STATCDI

COMPREHENSIVE DESCRIPTION AND PURPOSE

OVERALL

FUNCTION

STATCDI is the second of two program (EDITCDI is the first) created to work with 'Cruise Data Indices' (CDI'S) keypunched from the data logs of cruises sponsored by the Branch of Marine Geology, U.S.G.S., Menlo Park, CA. It performs two basic functions each of which compliment the functions of EDITCDI. First, it creates a 'Cruise Report' from a CDI which is both a condensed inventory of all the data listed in the CDI and a statistical summary of general cruise information and specific data information such as trackline distances and recording hours. Second, it can be used to do the final editing of a CDI by locating any remaining errors in the 'start/end/on/off' sequences for each data type of the CDI.

CRUISE REPORTS

Two versions of the 'Cruise Report' are possible: a 'Preliminary Report' and a 'Final Report'. The 'Preliminary' version can be generated immediately after a cruise before navigation for the cruise has been edited. It differs from the 'Final' version in that trackline distances for each data system are estimated by multiplying the average ship speed by the total recording hours for a given data system. Whereas, trackline distances for data systems in the 'Final' report are calculated directly from the navigation for the cruise by using trigonometric functions. Trackline distances for both versions are generally quite close but do differ for systems left on while the ship was dead-in-the-water or steaming at high speed on a transit line.

The reports are the same in every other respect. Both have a 'General Cruise Information' section which lists , with appropriate headers: area of ship operation; ship name; chief scientist(s); general types of data collected; times and places for port stops and the 'start/end' of the cruise; and the total underway and total port time. Each report also lists personnel, equipment used, and counts of each kind of sample, deployment, or underway data quantity collected by the cruise, along with total recording time for each type of underway data.

Information gleaned from this report is used by the Branch Chief and by individual scientists in their quarterly reports and open file reports on cruise data, etc.

PROGRAM STATCDI

EDITING

STATCDI is also used as the final editing step for a CDI before the CDI is run through the FINPRN subroutine of EDITCDI. The STAT subroutine of STATCDI keeps a strict account of any disruption of the 'start/end/on/off' sequence of a particular data type and will print the offending record's day/time and data type out to the user's terminal. It thus picks up any errors which may have been overlooked by visual editing.

An example of a 'Cruise Report' generated by STATCDI can be found in the Marine Data Information System documentation booklet.

STAT

FUNCTION

STAT is the entry program for STATCDI and carries the main burden of condensing information from a CDI for the output report. Its general function is to interactively receive information about the input data set, read the input navigation and CDI files, and then count each sample and data type encountered and tally total recording hours and trackline distances for each type of underway data.

PROGRAM STRUCTURE AND TASK

STAT operates on one CDI record at a time and two navigation records if navigation is being used. After reading a CDI record, it branches to one of seven sections depending on the data type of the record. One section deals with personnel, equipment, port and cruise records. The other sections handle essentially a single data category each; namely: geologic/water sample records, hydrographic records, observation/deployment records, station records, navigation records and geophysical records.

The section for personnel, equipment, port and cruise records, stores personnel and equipment names in output variables for the subroutine PRINT and links the on-board time for a person with his or her off-board time, so that the two times can be listed together. It also recognizes any difference in the spelling of each person's name enabling the subroutine PRINT to give the user the option of deleting an incorrect spelling. In addition, it adds up the total time at sea and the total port time. This is facilitated by converting the start and end times of cruise and port records to a continuous start/end/start/end..... sequence bracketing the at-sea intervals.

If station records are present, the station record section counts the number of unique stations and also calculates the total time on station for the cruise. Otherwise, this is done by the geologic/water sample section.

The section for observations or deployments tallies them and determines how many stations they were collected over.

The section for geologic and water samples counts the number of sample attempts and the number of sample recoveries for each sample type. It also tallies the sample recoveries by water depth interval and lists the total number for each interval and for all attempts and all recoveries. If there are no specific station records, the section will attempt to identify and count the different sampling stations occupied and total time on station by checking the record variable 'sp' which is supposed to record the station at which a sample was taken.

The sections for hydrographic, navigation, and geophysical records all make counts of the number of unique rolls, reels, or lists encountered but only the hydrographic and geophysical sections count the total recording time for each data system. The geophysical section performs the additional task of calculating trackline distances for each data system, provided navigation is available.

Other tasks common to 2 or more program sections are the identification of each 'specific' data type, the determination of 'start/end/on/off' sequences, and, if navigation is present, the calculation of cumulative trackline distances for the overall cruise and for each kind of underway data recorded.

DATA TYPE IDENTIFICATION

Except for the section dealing with station, equipment, personnel, port, and cruise records, each program section will try to identify a record more closely by searching a temporary storage array in which the 'specific' data type number codes (jc) of previously encountered records have been stored. This 'specific' data type numoer code is one of the 'additional variables' added to the record in AUTOSR of EDITCDI and relates a record's 4-letter 'Data Code' to a fixed position of the 'Data Code' in the program array 'cod'. The number is then used in place of the 'Data Code' as identification. If a record's number code is not located in the temporary storage array, it is added to this array. At the same time, the number code for the record's 'Media Code' (jm), will be stored in another array used for that purpose. This array is 2-dimensional and one of its elements will contain the position number of the record's 'specific' data type number code in its own array. Thus, a record's 'recording medium' or in some cases, its general data category, is linked to its 'specific' data type and it is this particular medium-data type combination which is statistically considered. An example might be gravity data which is stored both on magnetic tape and on paper rolls. The 'specific' data type is gravity, but the gravity is linked with both its magnetic tapes and its paper rolls so that trackline distances will be calculated for both gravity on magnetic tape and on paper rolls, and data quantities and recording hours for the two media will also be calculated separately.

PROPER 'START/END/ON/OFF' SEQUENCES

For hydrographic, geophysical, station, trackline and port/cruise records the 'start/end/on/off' sequence is checked. In general, whenever there is a 'start' record, the next record must be an 'end' or and 'off'; whenever there is an 'on' record the next record must be an 'off' or an 'end'; whenever there is an 'end' record, the next record must be a 'start'; and whenever there is an 'off' record, the next record must be an 'on'. If this sequence is disrupted, the program identifies the offending record to the user's terminal and if recording time is being calculated, the record will be used for accumulating time only if it is a 'start' or an 'on' record, otherwise a new record will be read.

OVERALL TRACKLINE DISTANCE

The overall distance covered by a cruise is trigonometrically calculated by calculating the distance between successive navigation records and CDI records which have navigation and adding them to a running total. If navigation has been merged with a CDI data file, every record except equipment and personnel records should have navigation positions. Each time a new non-personnel/equipment record is read, it will be operated on by its appropriate program section and then it will be used by the subroutine DIST to calcualte the distance between it and the previous record to have a navigational position. If navigation records fall between successive CDI records, the navigation records will be used to calculate the incremental distances along the ship's course.

When gaps are encountered in the navigation, an estimate is made of the distance over the gap by multiplying the time interval of the gap by an estimated averge speed of the ship which is based on the last 10 records to have navigation. If the ship occupied a station during a navigation gap, the station time is considered to be dead-in-the-water time and is subtracted out of the total time interval of the navigation gap before the time interval is multiplied by the average ship speed. This total overall trackline distance is used in the calculation of trackline distances for specific data systems.

PROGRAM STATCDI

TRACKLINE DISTANCES FOR UNDERWAY DATA SYSTEMS

Trackline distances are calculated for CDI trackline records and for all underway, continuously recorded geophysical data records. However, if navigation for the CDI is not available, trackline distance estimates are made in the subroutine PRINT by multiplying total trackline time and the recording times for each data system by an average ship speed.

When navigation is present, cumulative distances for each data system are determined by first storing the value of the 'cumulative overall' trackline distance at the time of each 'start' or 'on' record of a particular data system. Then, whenever an 'end' or an 'off' record is encountered for that data system, the value of the 'overall' trackline distance is again noted and the difference between the two values is taken and added to an array that stores the cumulative trackline distance for the data system. In effect, the trackline distances for each system are 'sectioned' out of the 'overall' trackline distance in proportion to the recording interval times of each system.

SUBROUTINES

There are five subroutines in STATCDI. DIST is called by STAT for each record with navigation; PRINT is called by STAT after every CDI file record has been read and acted on; SORT and DATCON are called from PRINT; and BLOCKDATA contains labeled Common and Data Statements used by PRINT.

DIST - Calculates Trackline Distances -

FUNCTIONS

DIST perfoms essentially three functions. First, it calculates trackline distance in kilometers between successive records which have navigation. This calculation is made using spherical trig functions taken from ____________________. 6368.7 KM is used as the mean diameter of the earth. Second, it calculates a moving average ship speed by averaging the speed over the 10 records previous to the one currently being operated on. This value is then used to estimate distances travelled during gaps in the navigation for a CDI file. An additional feature of this section is to check whether the ship speed as calculated from a currrent record is greater than 60 km/hr. If it is, the distance calculated from it will not be used to increment the 'overall' trackline distance and a new record will be read. Third, When a navigation gap is encountered, the average speed over the last 10 records is used to estimate the distance increment to be added to the 'overall' trackline distance value. This is done incrementally by multiplying the average speed by the time intervals between successive records within the gap. The navigation gap time interval is corrected by subtracting out any on-station time which may have occured during the gap, since when on-station, the ship is assumed to be relatively dead in the water.

PRINT - Prints Out Reports -

FUNCTIONS AND TASKS

PRINT organizes the data generated in STAT and DIST for output in a 'Preliminary' or 'Final' report format. There are ten possible sections to a report, each with its identifying header and in a fixed order. The data generated by each section is also put in a fixed order and if lists of data are being presented, the lists are hierarchally sorted in the subroutine SORT, so that similar types of data will be grouped together and so that different groups are always in the same relationship to one another. Also, for each section where lists of data will be printed out, the value of the 'specific' data type number code (jc) will be used to locate a verbal description of the data type stored in the Data Statement 'dout' (dot1, dot2, dot3) in the subroutine BLOCKDATA. The characteristics of each section are described below.

PROGRAM SECTIONS

1. The 'Cruise Report' header will indicate whether navigation was used for the report and whether the report is a 'Preliminary' or a Final' version. Also, there are two slightly different formats for the Cruise Locator: one is for locators which have only two characters in the Cruise ID field. The other is for Cruise ID's which have 3 or 4 characters.

2. General Cruise Information includes cruise area, shipname, chief scientists, general type of data collected, cruise dates, port stops, and total underway and port time. The user will be prompted for a specific description of the area of the cruise which he can enter at the terminal in 40 characters or less. The Cruise ID will be searched by the program for the ship code which corresponds to the ship codes in the Data Statement 'shp' located in BLOCKDATA. If one is found, its corresonding ship name located in 'ship' will be printed out, otherwise the user will be prompted for the proper shipname. The

user will also be prompted for the name or names of the chief scientist(s) which he can enter in 32 or less characters. The general data type is predetermined in STAT and is automatically printed out. Times for cruise dates and port stops are converted to local dates and times by the subroutine DATCON, and port stops will be numbered. Total underway time and total port time is expressed in hours and in days and hours.

3. The Personnel section prints out a numbered list of all names logged only once to the user's terminal. A name which is logged twice but is misspelled, will appear on this list. The user will then be asked how many pairs of duplicate names there are or rather how many pairs of names have been misspelled, and then will be asked to enter the numbers of each duplicate name for each pair using an (i2,1x,i2) entry format. Entries will be prompted by >>, and the number of the name to be deleted should be entered first. The correct name will be used in the report while the 'on' or 'off' time of the incorrect name will be combined with that of the correct name.

4. The Equipment section rearranges the array containing the equipment list as transmitted by STAT into an array where every 4th element is the same data category. This enables the program to print out 4 columns, where each column is of the same general equipment category.

5. The Geophysical section calculates trackline nautical miles from kilometers for each system and prints both values out along with the number of recording hours and the number of rolls, or reels, etc., used. An additional feature is to print information for data from a single system but recorded on different media, together.

6. The Geologic/Hydrologic sample section prints the number of sample attempts and recoveries for each sample type. It also prints the number of sample recoveries in the depth intervals of 0-100m, 100-3000m, and >3000m, for each sample type. Finally, it adds up the total number of sample attempts and the total number of recoveries for both the overall category and for each depth interval.

7. The Geologic/Hydrologic underway data section is the same as the Geophysical section but does not print out trackline distances.

8. The Numerical Observation section prints out the number of observations taken and the number of stations they were taken over.

9. Navigation data is printed out in the same manner as the underway Geologic/Hydrologic data but does not include recording hours.

10. The operations section prints out the number of tracklines run, total trackline time, cumulative trackline distance; total time on station and number of stations occupied; and the number and type of deployments.

SORT - Sorts Data Lists -

Sort is called up by each program section in the subroutine PRINT which prints out lists of specific data types. Two variables are passed to SORT by PRINT, a variable containing the hierarchal numbers of each 'specific' data type and the number of data types being dealt with. SORT then arranges the hierarchal numbers in ascending order and passes their new positions back to PRINT in an index variable, 'nj'.

DATCON - Converts Julian Day/GMT to Local Standard Time -

DATCON is called up by the program section in PRINT that deals with 'General Cruise Information'. It is used specifically to convert the dates and times for the beginning and end of a cruise and for port stops into local standard dates and times. A 'modulo' function separates out the start-of-cruise or port-stay entries, and for each of these, the user will be prompted to enter the port's longitude in decimal degrees. The longitude is then divided by 15 degrees to determine how many time zones away from Greenwich Meridian the port is in order to give an adjustment factor used to convert GMT to local standard times.

Next, an estimate of the month is made by dividing the 'Julian Day' of the year by 30.42. Further tests of this estimate give the exact month of the year after leap years are taken into account. Finally, the exact day of the month is determined by subtracting the Julian day from the Julian day of the last day of the determined month which is stored in the Data Statements 'edmo' or 'edmol', depending on whether it is a leap year or not.

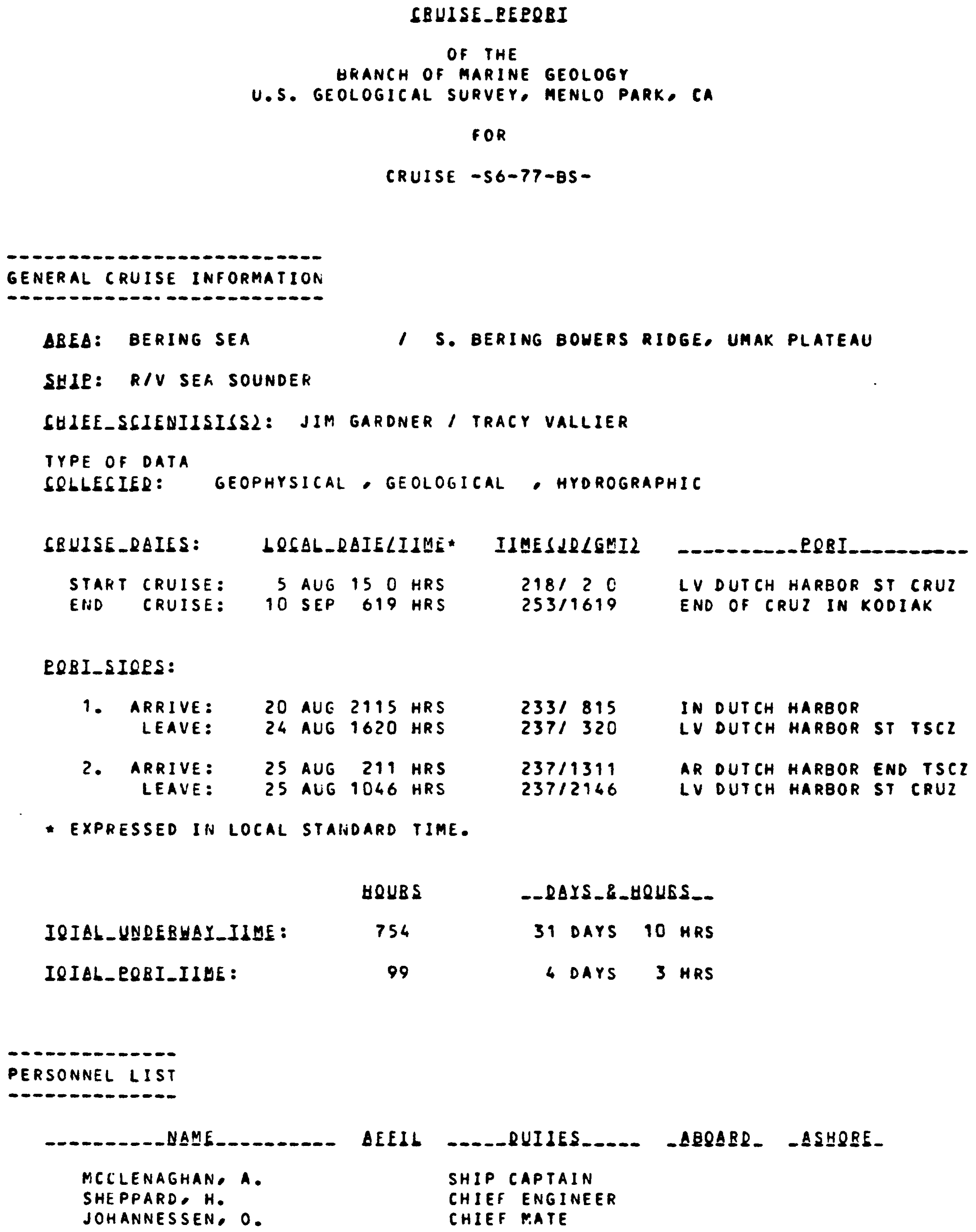
BLOCKDATA - Block Data Subroutine -

BLOCKDATA is a block data subroutine which stores literal values in Data Statements for variables which are used in the subroutine PRINT. The variables are transmitted in a

PROGRAM STATCDI

Labeled Common statment called 'block'.

There are three groups of Data Statements. The first contains variables holding area codes, area descriptions, blanks, deployed systems, months, underscores, ship codes, ship names, and the names of the 4 general data categories. The second contains variables holding media and general data group descriptors. The third contains the variables 'dot1', 'dot2', and 'dot3' which are equivalent to 'dout' and which hold the 'specific' data type descriptors which are linked to 'Data Codes' by 'jc' or 'ic'.



```
                                 CRUISE_REPORT

                                    OF THE
                           BRANCH OF MARINE GEOLOGY
                    U.S. GEOLOGICAL SURVEY, MENLO PARK, CA

                                      FOR

                               CRUISE -S6-77-BS-


--------------------------
GENERAL CRUISE INFORMATION
--------------------------

   AREA:  BERING SEA            /  S. BERING BOWERS RIDGE, UMAK PLATEAU

   SHIP:  R/V SEA SOUNDER

   CHIEF_SCIENTIST(S):  JIM GARDNER / TRACY VALLIER

   TYPE OF DATA
   COLLECTED:     GEOPHYSICAL , GEOLOGICAL   , HYDROGRAPHIC

   CRUISE_DATES:     LOCAL_DATE/TIME*   TIME(JD/GMT)     ----------PORT---------

     START CRUISE:     5 AUG 15 0 HRS      218/ 2 0      LV DUTCH HARBOR ST CRUZ
     END   CRUISE:    10 SEP  619 HRS      253/1619      END OF CRUZ IN KODIAK


   PORT_STOPS:

      1.  ARRIVE:     20 AUG 2115 HRS      233/ 815      IN DUTCH HARBOR
             LEAVE:   24 AUG 1620 HRS      237/ 320      LV DUTCH HARBOR ST TSCZ

      2.  ARRIVE:     25 AUG  211 HRS      237/1311      AR DUTCH HARBOR END TSCZ
             LEAVE:   25 AUG 1046 HRS      237/2146      LV DUTCH HARBOR ST CRUZ

   * EXPRESSED IN LOCAL STANDARD TIME.


                             HOURS        --DAYS_&_HOURS--

   TOTAL_UNDERWAY_TIME:       754          31 DAYS  10 HRS

   TOTAL_PORT_TIME:            99           4 DAYS   3 HRS


-------------
PERSONNEL LIST
-------------

   ----------NAME----------  AFFIL  -----DUTIES-----  _ABOARD_  _ASHORE_

      MCCLENAGHAN, A.                SHIP CAPTAIN
      SHEPPARD, H.                   CHIEF ENGINEER
      JOHANNESSEN, O.                CHIEF MATE
```

FIGURE 27   EXAMPLE OF A 'CRUISE REPORT' GENERATED FROM A CRUISE DATA INDEX FOR CRUISE S6-77-BS

```
GARDNER, JAMES          UGS   CHIEF SCIENTST    218/ 2 0  253/17 0
VALLIER, TRACY          UGS   CHIEF SCIENTST    218/ 2 0  253/17 0
DEAN, WALTER            UGS   GEOLOGIST         218/ 2 0  253/17 0
STANLEY, EDITH          UCD   GEOLOGIST         218/ 2 0  253/17 0
BUDAI, ALEXIS           UGS   GEOLOGIST         218/ 2 0  253/17 0
GOUD, MARGARET          UGS   GEOLOGIST         218/ 2 0  253/17 0
REED, DON               UGS   GEOLOGIST         218/ 2 0  253/17 0
UNDERWOOD, MIKE         UGS   GEOLOGIST         218/ 2 0  253/17 0
LEWIS, STEVE                  GEOLOGIST         218/ 2 0  235/ 2 0
BARNES, NEAL                  GEOLOGIST         218/ 2 0  253/17 0
KARL, HERMAN                  GEOLOGIST         218/ 2 0  253/17 0
FROST, TOM                    GEOLOGIST         235/ 2 0  253/17 0
HESS, GORDON                  GEOLOGIST         235/ 2 0  253/17 0
KLISE, DAVID            UGS   GEOLOGIST         218/ 2 0  253/17 0
BRADY, ROLAND           UGS   GEOLOGIST         234/ 2 0  253/17 0
KVENVOLDEN, K.          UGS   GEOCHEMIST        218/ 2 0  235/ 2 0
REDDEN, GEORGE          UGS   GEOCHEMIST        218/ 2 0  253/17 0
CUDNOHUFSKY, J.               BIOLOGIST         235/ 2 0  253/17 0
PATRICK, R.             UGS   ELECTRONICS T     218/ 2 0  235/ 2 0
SALADIN, JOHN           UGS   ELECTRONICS T     234/ 2 0  253/17 0
HARTMAN, TOM                  MECHANICAL T      234/ 2 0  237/21 0
WINEBRENNER, C.               MECHANICAL T      234/ 2 0  237/21 0
GARLOW, RICHARD         UGS   NAVIGATOR         218/ 2 0  235/ 2 0
GIBBONS, HELEN          UGS   NAVIGATOR         218/ 2 0  253/17 0


---------------------
EQUIPMENT SYSTEMS USED
---------------------

   --NAVIGATIONAL--    --GEOPHYSICAL---     ---GEOLOGICAL---    -HYDROGRAPHICAL-

   INTEGRATED NAV      SNGL CHANL ARCER     PISTON CORE         TRANSMISSOMETER
   NAV SATELLITE       MINISPARKER          GRAVITY CORE        HYD CURRENT METR
   DOPPLER SONAR       UNIBOOM              VAN VEEN GRAB       CTD METER
   LORAN C             3.5KH BATHYMETRY     CHAIN DREDGE        EXP BATHY THERMO
                       12KH BATHYMETRY      SEAFLOOR CAMERA     TEMP/SALINOMETER
                       SHIPBOARD MAGGY      TELEVISION          VAN DORN BOTTLE
                       SHIPBOARD GRAVTY                         NISKEN BOTTLE
                       SIDE SCAN SONAR


--------------
DATA COLLECTED
--------------

   GEOPHYSICAL

      DATA TYPE         RECORDING      TRACKLINE TRACKLINE RECORDING ROLL,REEL
   ---OR-SYSTEM----  ----MEDIUM----  KILOMETRS N.-MILES- TIME(HRS) LIST-QTY-

   SNGL CHANL ARCER  ANL PAPER ROLL   7233.6    3905.9     504.8       14

   UNIBOOM           ANL PAPER ROLL    168.1      90.8      16.8        3

   MINISPARKER       ANL PAPER ROLL     58.4      31.6       4.4        2

   3.5KH BATHYMETRY  ANL PAPER ROLL   8566.5    4625.5     725.0       23
```

FIGURE 27 (CONTINUED)

```
  12KM BATHYMETRY   ANL PAPER ROLL  6859.5    3703.9      606.1       15

  SHIPBOARD MAGGY   ANL PAPER ROLL  4524.8    2443.2      313.3        7

  MAG/BATHYM/NAVIG  PRINTR LISTING  8364.5    4516.4      724.6        5
                    DIGIT MAG TAPE  8726.2    4711.8      749.2       20

  SHIPBOARD GRAVTY  ANL PAPER ROLL  8381.3    4525.5      709.6       21
                    PRINTR LISTING  8810.4    4757.2      846.1        3
                    DIGIT MAG TAPE  8369.4    4519.1      708.6       43

GEOLOGICAL/HYDROLOGICAL_SAMPLES
                                                NUMBER OF SAMPLES FROM A
   SAMPLING           SAMPLING  SAMPLES       GIVEN WATER DEPTH INTERVAL
   DEVICE__________   ATTEMPTS_ RECOVERED    _0-100M__ 100-3000M __>3000M_

   GRAVITY CORE          29        28           0        26         2
   VAN VEEN GRAB          7         4           2         2         0
   CHAIN DREDGE           5         4           1         3         0
   VAN DORN BOTTLE      147       143           0       143         0
   EXP BATHY THERMO      13        12           0        11         1
   CTD METER             41        41           0        41         0
                     ------    ------      ------    ------    ------
   TOTALS               242       232           3       226         3

GEOLOGICAL/HYDROLOGICAL_(ANALOG)

      DATA TYPE          RECORDING          RECORDING   NUMBER OF TAPES,
   ___OR_SYSTEM____   ____MEDIUM____        TIME(HRS)   ROLLS,LISTS,ETC.

   TELEVISION         ANLOG MAG TAPE           5.8             2

   SEAFLOOR CAMERA       PHOTOGRAPH            0.0             8

   HYD CURRENT METR   ANL PAPER ROLL         133.0             1

   TEMP/SALINOMETER   ANL PAPER ROLL         684.5             5

   CTD METER          ANL PAPER ROLL         138.8             1
                      DIGIT MAG TAPE         140.2             2

NUMERICAL_OBSERVATION_

              DATA TYPE         NUMBER OF    TAKEN OVER HOW
           ___OR_SYSTEM____     READINGS_    MANY_STATIONS_

           HYD CURRENT METR        27             27


----------------------
OPERATIONS INFORMATION
----------------------

   STATION_DATA    STATIONS OCCUPIED:  64,   TOTAL TIME ON STATION: 133.2 HRS,

   TRACKLINES          TRACKLINES RUN: 152,  TOTAL TRACKLINE TIME:  625.8 HRS,

            CUMULATIVE TRACKLINE DISTANCE:  8407.5 KM / 4539.7 N. MILES
```

FIGURE 27 (CONTINUED)

```
COMPILATION LISTING OF SEGMENT stat
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1651.3 pst Tue
    Options: table symbols

c********* STATCDI MAINLINE *********
      logical lap,lbadnv,ldis,lend,lfg,lfh,lfst,lgap,lgel,lgep,lhyd,
      lmarc,lmgvx,lnm,lnu,lnv,lsoc,lpgp,lps,lps9,lsad,lsam,lshnt,lso,
      lst,lstat,lstint,lstp,ldx
c
      double precision dam1,dam2,dlat1,dlat2,dlon1,dlon2,dsdlat,dsdlon,
      dcse,dspd,dumya8
c
      character*1 iu,slave,un
      character*4 ibc,idc,idd,isn,idp,isnm,paf,sp,stan
      character*8 ibk,ordnl(7)
      character*32 segin,segout
c
      integer dsam,dy,hirg,hirh,hirp,hirq,hirx,hr,ib,l1,pjc,rec,s
c
      dimension cuvt(30,10),dxmrk(30,10),ise(30,10),isnm(10,600),
      iso(30,10),ker(30,10),kerr(30,10),l1(5),lfg(30,10),lfh(30,10),
      tbnv1(30,10),tmg1(30,10),tmh1(30,10),dfn(100),dln(100),idd(6),
      stan(600),tes(100),tfn(100),tln(100),tmc1(30),tmc2(30),tss(100)
c
c ---- LABELED COMMON STATEMENTS ----
      common/dis/rltn1,rlnn1,rmnv1,tsst,test,tmc,dxc,dx,istint,nvd1,nvh1
      ,nv,nw,lbadnv,ldis,lfst,lps9,lsam,lshnt,lst,lstint
      common/stprdt/idc(30,6),jdc(30),ihc(30),kc,s(20)
      common/stpr/ibk,cucrz,cugeo(30,10),cuhyd(30,10),cuprt,custa,dsam(3%
      0,3),paf(70),rmc(30),rmp(70,2),rnot,sysdx(30,10),hirg(30),hirh(30)
      ,hirp(70),hirq(60),hirx(30),ib,idat(30),ide(10),idp(70,6),ihyd(20),
      ihir(10),ihp(70,2),imed(30,10),inav(30),iob(10),iquip(60),isam(30),
      itrk,un,jdp(70,2),jmk(30),jml(10),jmn(10),kdep(10),keq,khym(30,10),
      kmd(30,10),kmed(30,10),kn,knm(30,10),kob(10),kp,kper(70),ksam,
      ksn(10),ksta,medk(30,10),n,nd,nh,no,nrec(30),nsam(30),pjc(70),lgep,
      lgel,lhyd,lnv,lsad,lstat
c
c ---- DATA STATEMENTS INITIALIZING VARIABLES ----
      data ordnl/"FIRST   ","SECOND  ","THIRD   ","FOURTH  ","FIFTH   ",
      "SIXTH   ","SEVENTH "/,n,kc,kn,kp,nh,keq,no,ksam,ksta,cucrz,cumdx,
      cuprt,custa,itrk,istc,tnv1,tnv2,nv,nw,dxc,tmc,tsst,test,istint,nd,
      istn/25*0,2/,rec,ib,ibc,ibk/"Y","    ","    ","        "/,lap,lend,
      lgap,lgel,lgep,lhyd,lnu,lps,lps9,lpgp,lshnt,lst,lstp,
      lstint/14*.false./,l1/"X","G","V","H"," "/,slave/"S"/
c
c ---- INITIALIZE VARIABLES ----
      nr=0
      do 6 i=1,30
      jmk(i)=0
      nrec(i)=0
      nsam(i)=0
      dsam(i,1)=0
      dsam(i,2)=0
      dsam(i,3)=0
      jdp(i,1)=0
      jdp(i,2)=0
      kper(i)=0
      na=i+30
      jdp(na,1)=0
```

```
            jdp(na,2)=0
            kper(na)=0
            do 6 j=1,10
            cugeo(i,j)=0
            cuhyd(i,j)=0
            cuvt(i,j)=0
            ker(i,j)=0
            kerr(i,j)=0
            khym(i,j)=0
            knm(i,j)=0
            medk(i,j)=0
            tbnv1(i,j)=0
            tmg1(i,j)=0
            tmh1(i,j)=0
 6          continue
            do 10 i=1,10
            jml(i)=0
            jmn(i)=0
            kdep(i)=0
            kob(i)=0
            na=i+60
            jdp(na,1)=0
            jdp(na,2)=0
            kper(na)=0
 10         ksn(i)=0
 c
 c ---- PROMPTS FOR SPECIFIC INFORMATION ABOUT INPUT CDI AND OUTPUT REPORT
 c         NAME ----
            write(6,12)
 12         format(" ??WERE STATIONS ENTERED IN 'sp'?? (t OR f)(l1)")
            read(5,26) lsad
            write(6,14)
 14         format(" ??WERE STATIONS ENTERED AS STAT RECORDS?? (t OR f)(l1)")
            read(5,26) lstat
            ldis=.true.
            write(6,16)
 16         format(" ENTER INPUT SEGMENT NAME - UP TO 32 CHARACTERS.")
            read(5,18) segin
 18         format(a32)
            call io ("attach","file08","vfile_",segin)
            call io ("open","file08","si")
            write(6,20)
 20         format(" ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.")
            read(5,18) segout
            call io ("attach","file42","vfile_",segout)
            call io ("open","file42","so")
            write(6,22)
 22         format(" ??WILL THIS PROGRAM-RUN USE NAVIGATION FILES?? (t OR f)",%
            "(l1)")
            read(5,26) lnv
 26         format(l1)
            write(6,28)
 28         format(" ??HAS NAVIGATION BEEN ADDED TO CDI?? (t OR f)(l1)")
            read(5,26) lnm
            if(lnv) go to 34
            write(6,30)
 30         format(" ENTER AVERAGE SHIP SPEED FOR CRUISE. (f4.1 - EG: 05.5)")
            read(5,32) rnot
 32         format(f4.1)
            go to 60
```

```
34         write(6,36)
36         format(" IF NAV HAS NOT BEEN MERGED w/CDI, PROGRAM TERMINATES.")
           if(.not.lnm) go to 1600
           write(6,38)
38         format(" ??IS MAG. INPUT ON MAGNAVOX TAPE?? (t OR f)(l1)")
           read(5,26) lmgvx
           if(lmgvx) go to 46
           write(6,40)
40         format(" ??IS MAG. INPUT ON MARCONI TAPE?? (t or f)(l1)")
           read(5,26) lmarc
           if(lmarc) go to 52
c
c
c********* READ FIRST RECORDS AND BRANCH TO APPROPRIATE DATA SECTION *********
c
c ---- READ IN FIRST TWO NAV. RECORDS ----
           read(10,44) nvd1,nvh1,rmnv1,rltn1,rlnn1
c ---- CONVERTS JULIAN DAY AND DAY/TIME OF 1ST NAV RECORD TO MINUTES AND STORES
c         IN tnv1 ----
           tnv1=1440*nvd1+60*nvh1+rmnv1
42         read(10,44,end=60) nvd2,nvh2,rmnv2,rltn2,rlnn2
44         format(i3,1x,i2,f4.1,44x,f7.5,1x,f9.5)
           go to 60
c ---- READS NAV FROM MAGNOVOX TAPE ----
46         read(10) nvd1,nvh1,rmnv1,rltn1,rlnn1
           tnv1=1440*nvd1+60*nvh1+rmnv1
48         read(10,end=60) nvd2,nvh2,rmnv2,rltn2,rlnn2
           go to 60
c ---- READS NAV FROM MARCONI TAPE ----
52         read(10) nvd1,nvh1,dam1,iline,ishot,itype,dlat1,dsdlat,dlon1,
           dsdlon,dcse,dspd,dumya8
           rmnv1=sngl(dam1)
           rltn1=sngl(dlat1)
           rlnn1=sngl(dlon1)
56         read(10,end=60) nvd2,nvh2,dam2,iline,ishot,itype,dlat2,dsdlat,
           dlon2,dsdlon,dcse,dspd,dumya8
           rmnv2=sngl(dam2)
           rltn2=sngl(dlat2)
           rlnn2=sngl(dlon2)
c
c ---- READ IN FIRST CDI RECORD ----
60         if(.not.lnm) go to 74
           read(8,71,end=1500) dy,hr,rmn,lin,sp,isn,(idd(l),l=1,6),iu,
           (s(l),l=1,20),jc,jm,ilt,iln
71         format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
           t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
           go to 80
74         read(8,77,end=1500) dy,hr,rmn,lin,sp,isn,(idd(l),l=1,6),iu,
           (s(l),l=1,20),jc,jm
77         format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
           t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3)
80         lsoc=.false.
c
c ---- SETS LOGICAL VARIABLES lgep,lgel,lhyd = true IF GEOPHYSICAL,
c         GEOLOGIC, OR HYDROGRAPHIC RECORDS EXIST IN THE INPUT CDI DATA SET ----
           if(lgep.and.lgel.and.lhyd) go to 98
           do 83 i=1,4
           if(s(16).eq.l1(i)) go to 86
83         continue
86         go to (89,92,98,95),i
```

```
89          if(lgep) go to 98
            lgep=.true.
            go to 98
92          if(lgel) go to 98
            lgel=.true.
            go to 98
95          lhyd=.true.
98          hir=s(20)
            nn=3
c ---- lsam = false => CURRENT RECORD NOT A SAMPLE, UNTIL PROVEN OTHERWISE ----
            lsam=.false.
c
c ---- TRANSLATE ST/ED/ON/OFF TO ista=1/2/3/4 ----
            do 100 m=1,4
            nn=nn+1
            if(s(nn).ne.ib) go to 120
100         continue
            ista=0
            go to 130
120         ista=m
c
c ---- SENDS RECORDS W/'NAV','PER','EQP' MEDIA CODES TO STATEMENT 400 ----
130         if(hir.lt.200000) go to 420
c ---- SENDS SAMPLES TO 135 ----
            if(jm.eq.4) go to 135
c ---- SENDS OBSERVATION RECORDS TO 372 ----
            if(jm.eq.8) go to 372
c ---- SENDS DEPLOYMENT RECORDS TO 404 ----
            if(jm.eq.12) go to 404
c ---- SENDS GEOPHYSICAL RECORDS TO 600, LEAVES GEOLOGIC & HYDROG. RECORDS ----
            if(hir.lt.300000) go to 600
c ---- SENDS NON-SAMPLES (MOSTLY HYDROG.) TO STATEMENT 280 ----
            go to 280
c
c
c********* GEOLOGIC DATA PLUS WATER SAMPLES *********
135         lsam=.true.
            if(ksam.eq.0) go to 160
c
c ---- SEARCHES FOR SAMPLE TYPE IN STORAGE VARIABLE, 'isam' ----
            do 140 i=1,ksam
            if(jc.eq.isam(i)) go to 170
140         continue
160         ksam=ksam+1
c ---- STORES SAMPLE TYPE IN 'isam' ----
            un=iu
            isam(ksam)=jc
            hirg(ksam)=s(20)
            i=ksam
170         if(ista.eq.1) go to 255
c ---- SEPARATES RECOVERIES FROM NO-RECOVERIES ----
            if(s(8).eq.rec) go to 200
c ---- KEEPS COUNT OF EACH SAMPLE TYPE ----
            nrec(i)=nrec(i)+1
            go to 255
c ---- COUNTS RECOVERIES FOR PARTICULAR SAMPLE TYPE ----
200         nsam(i)=nsam(i)+1
            idep=0
c
c ---- CALCULATES SAMPLE DEPTH ----
```

```
           do 220 k=9,12
           kk=12-k
220        idep=s(k)*10**kk+idep
c ---- CALCULATES SAMPLE DEPTH-RANGE & KEEPS COUNT OF NUMBER OF SAMPLES IN A
c        PARTICULAR DEPTH-RANGE FOR A PARTICULAR SAMPLE TYPE ----
           if(idep.gt.2999) go to 240
           if(idep.lt.100) go to 250
           dsam(i,2)=dsam(i,2)+1
           go to 255
240        dsam(i,3)=dsam(i,3)+1
           go to 255
250        dsam(i,1)=dsam(i,1)+1
c ---- lst OR lstat = true => STATION RECORDS EXIST FOR THIS SAMPLE; THEREFORE,
c        SAMPLES WON'T BE USED TO DETERMINE STATION NO.S ----
255        if(.not.lsad) go to 276
           if(lstat) go to 276
           if(lst) go to 276
           if(sp.ne.ibc) go to 257
           write(6,259) s(2),dy,hr,rmn
259        format(" STATION NO. NOT ENTERED IN -SP- FOR SAMPLE ",a4," OF DA",%
           "Y/TIME ",i3,"/",i2,f4.1)
           go to 276
257        if(ksta.eq.0) go to 268
c ---- TRYING TO LOCATE CURRENT RECORD'S STATION NO. IN ARRAY OF PREVIOUSLY
c        ENCOUNTERED STATION NO.'S ----
           do 264 i=1,ksta
           if(sp.eq.stan(i)) go to 272
264        continue
c ---- COUNT OF UNIQUE STATION NUMBERS ENCOUNTERED , INCREMENTED BY 1 ----
268        ksta=ksta+1
c ---- UNIQUE STATION NUMBER STORED IN 'stan' ----
           stan(ksta)=sp
c ---- lfst = true => CURRENT RECORD IS FIRST SAMPLE ENCOUNTERED ON THIS STATION
c        AS SIGNALLED BY RECORD VARIABLE 'sp' ----
           lfst=.true.
c ---- tsst AND test HOLD START AND END TIMES FOR A PARTICULAR STATION ----
c ---- THE 'START/END TIMES' OF A STATION CALCULATED FROM 'sp', ARE CALCULATED
c        FOR EACH SAMPLE RECORD FROM THAT STATION.  ADDED TOGETHER, THEY MAKE
c        GOOD ESTIMATES OF TOTAL STATION TIME AND FIT THE LOGIC OF THE PROGRAM ----
           tsst=1440*dy+60*hr+rmn-10.
           test=tsst+10.
           custa=custa+20.
           go to 276
272        lfst=.false.
           tsst=test
           test=1440*dy+60*hr+rmn
           custa=custa+test-tsst
276        if(.not.lnv) go to 60
c ---- lnu = true => A RECORD IS NOT UNDERWAY DATA.  BUT, IT WILL BE SENT THRU
c        CUMULATIVE DX ALGORITHM TO ACT AS NAVIGATION RECORD ----
           lnu=.true.
           go to 850
c
c
c********* HYDROGRAPHIC DATA (MOSTLY) *********
280        if(nh.eq.0) go to 304
           nn=nh
c
c ---- TRIES TO MATCH RECORD'S DATA TYPE TO DATA TYPES PREVIOUSLY FOUND AND
c        STORED IN 'ihyd' ----
```

```
          do 29o i=1,nn
          if(jc.ne.ihyd(i)) go to 296
          kk=jml(i)
c ---- TRIES TO MATCH RECORD'S MEDIA TYPE TO MEDIA TYPES PREVIOUSLY FOUND AND
c         STORED IN 'kmed' ----
          do 288 k=1,kk
          if(jm.ne.kmed(i,k)) go to 288
          if(.not.lfh(i,k)) go to 320
          l=k
          go to 314
  288     continue
          go to 312
  296     continue
c ---- COUNTS UNIQUE DATA CODES ENCOUNTERED ----
  304     nh=nh+1
c ---- STORES DATA TYPES ENCOUNTERED FOR FIRST TIME IN 'ihyd' ----
          ihyd(nh)=jc
          hirh(nh)=s(20)
          i=nh
c ---- COUNTS UNIQUE MEDIA TYPES ENCOUNTERED FOR EACH DATA TYPE ----
  312     jml(i)=jml(i)+1
          l=jml(i)
c ---- STORES MEDIA TYPES FOUND WITH A PARTICULAR DATA TYPE FOR THE FIRST TIME,
c         IN 'KMED' ----
          kmed(i,l)=jm
c ---- lfh = t INDICATES SEARCHING FOR PROPER 'ST/ON' SEQUENCE FOR 1ST
c         RECORD OF A PARTICULAR MEDIA/DATA TYPE COMBINATION ----
          lfh(i,l)=.true.
c ---- CHECKS WHETHER 1ST RECORD FOR A PARTICULAR MEDIA/DATA TYPE COMBINATION IS
c         A 'START' OR 'ON'; IF IT IS, SHUNTS TO STATEMENT 356 ----
  314     if(ista.eq.1.or.ista.eq.3) go to 356
c ---- COUNT OF DISRUPTIONS TO PROPER 'ST/END/ON/OFF' SEQUENCE FOR EACH MEDIA/
c         DATA TYPE COMBINATION ----
          if(ista.ne.0) go to 315
          if(jm.eq.10.or.jm.eq.13) go to 323
c ---- COUNTS ERRORS IN ST/ED/ON/OFF SEQUENCE ----
  315     ker(i,l)=ker(i,l)+1
          write(6,316) ordnl(ker(i,l)),s(1),s(2)
  316     format("0",a8,1x,"RECORD OF DATA TYPE -",a3,1x,a4,"- IS NOT A ST",%
          "ART/ON. TIME ACCUMULATION WILL NOT BEGIN UNTIL A START/ON.RECOR",%
          "D IS ENCOUNTERED.")
          write(6,318) ordnl(ker(i,l)),dy,hr,rmn
  318     format(" DAY/TIME FOR THE ",a8," RECORD IS: ",i3,"/",i2,f4.1)
          go to 60
  320     l=k
c ---- BRANCHES, DEPENDING ON WHETHER PREVIOUS RECORD WAS 'ST/END/ON/OFF' ----
          ii=iso(i,l)
          go to (321,336,321,344), ii
c ---- SENDS 'END/OFF' RECORDS TO STATEMENT 352 ----
  321     if(ista.eq.2.or.ista.eq.4) go to 352
          if(ista.ne.0) go to 327
  322     if(jm.ne.10.or.jm.ne.13) go to 325
c ---- khym COUNTS UNIQUE ROLLS, REELS, LISTS, ETC. ----
  323     khym(i,l)=khym(i,l)+1
          iso(i,l)=2
          go to 60
  325     write(6,726) s(1),s(2),dy,hr,rmn
          iso(i,l)=2
          ker(i,l)=ker(i,l)+1
          go to 60
```

```
327       ker(i,l)=ker(i,l)+1
          write(6,330) s(1),s(2),dy,hr,rmn
330       format("ORECORD OF DATA TYPE -",a3,1x,a4,"- AND DAY/TIME ",i3,"/",
          i2,f4.1," BREAKS PROPER ST/ED/ON/OFF SEQUENCE.")
          write(6,332)
332       format("    IT WILL BE USED AS NEW STARTING POINT FOR ACCUMULATI",%
          "NG TIME.")
          go to 360
c ---- SENDS 'START' RECORDS TO STATEMENT 360 ----
336       if(ista.eq.1) go to 360
          if(ista.eq.0) go to 322
          ker(i,l)=ker(i,l)+1
          write(6,330) s(1),s(2),dy,hr,rmn
          write(6,340)
340       format(" IF RECORD IS AN -ON-, IT WILL BE USED AS STARTING POINT",%
          " FOR ACCUMULATING TIME;"/"  OTHERWISE, A NEW RECORD WILL BE REA",%
          "D.")
          if(ista.eq.3) go to 364
          go to 60
c ---- SENDS 'ON' RECORDS TO STATEMENT 364 ----
344       if(ista.eq.3) go to 364
          if(ista.eq.0) go to 322
          ker(i,l)=ker(i,l)+1
          write(6,330) s(1),s(2),dy,hr,rmn
          write(6,346)
346       format(" IF RECORD IS A -START-, IT WILL BE USED AS STARTING POI",%
          "NT FOR ACCUMULATING TIME;"/"  OTHERWISE, A NEW RECORD WILL BE R",%
          "EAD.")
          if(ista.eq.1) go to 360
          go to 60
c ---- CALCULATES TIME IN HOURS FOR 'END/OFF' RECORDS, AND STORES IN 'tmh2' ----
352       tmh2=1440*dy+60*hr+rmn
c ---- CALCULATES TIME DIFFERENCE BETWEEN 'ST/ON' AND 'END/OFF' RECORDS & ADDS
c        TO CUMULATIVE TIME VARIABLE 'cuhyd' FOR EACH MEDIA/DATA TYPE COMBINA-
c        TION ----
          cuhyd(i,l)=cuhyd(i,l)+(tmh2-tmh1(i,l))
          go to 368
356       lfh(i,l)=.false.
c ---- khym COUNTS # OF UNIQUE ROLLS, REELS, LISTS ETC. FOR EACH DATA TYPE ----
360       if(ista.eq.1) khym(i,l)=khym(i,l)+1
364       tmh1(i,l)=1440*dy+60*hr+rmn
368       iso(i,l)=ista
          if(.not.lnv) go to 60
          lnu=.true.
          go to 850
c
c
c********* OBSERVATIONAL DATA *********
372       if(no.eq.0) go to 384
c ---- TRIES TO MATCH CURRENT RECORD'S OBS/DATA TYPE TO THOSE PREVIOUSLY STORED
c        IN 'iob' ----
          if(ista.gt.1) go to 255
          do 378 i=1,no
          if(jc.ne.iob(i)) go to 378
          if(.not.lsad) go to 398
          kk=ksn(i)
c ---- TRIES TO MATCH CURRENT RECORD'S STATION NO.S TO THOSE PREVIOUSLY STORED
c        IN 'isnm' ----
          do 376 k=1,kk
          if(isnm(i,k).eq.sp) go to 398
```

```
376       continue
          go to 390
378       continue
c ---- COUNT OF UNIQUE 'OBS/DATA' TYPE COMBINATIONS ----
384       no=no+1
          i=no
c ---- STORES UNIQUE 'OBS/DATA' TYPE IN 'iob' ----
          iob(i)=jc
c ---- STORES UNIQUE HIERARCHAL # FOR PARTICULAR 'OBS/DATA' TYPE IN 'ihir' ----
          ihir(i)=hir
          if(.not.lsad) go to 398
c ---- COUNTS # OF UNIQUE STATION #'S PER UNIQUE 'OBS/DATA' TYPE ----
390       ksn(i)=ksn(i)+1
          k=ksn(i)
          if(sp.ne.ibc) go to 396
          write(6,392) s(1),s(2),dy,hr,rmn
392       format("0STATION FIELD FOR RECORD OF DATA TYPE -",a3,1x,a4,"- AN",%
          "D DAY/TIME OF -",i3,"/",i2,f4.1,"- IS BLANK."/"     ENTER PROPER",%
          " STATION #. (a4)")
          read(5,394) sp
394       format(a4)
c ---- STORES UNIQUE STATION #'S FOR UNIQUE 'OBS/DATA' TYPES IN 'isnm' ----
396       isnm(i,k)=sp
c ---- COUNTS # OF UNIQUE OBSERVATIONS PER DATA TYPE ----
398       kob(i)=kob(i)+1
          go to 255
c
c********* DEPLOYMENTS *********
404       if(nd.eq.0) go to 412
          if(ista.gt.1) go to 255
c ---- TRIES TO MATCH RECORD'S DATA TYPE TO THOSE PREVIOUSLY STORED IN
c        'ide' ----
          do 408 i=1,nd
          if(jc.eq.ide(i)) go to 416
408       continue
412       nd=nd+1
          i=nd
c ---- STORES RECORD'S DATA TYPE IN 'ide' ----
          ide(i)=jc
c ---- COUNTS NUMBER OF DEPLOYMENTS PER DATA TYPE ----
416       kdep(i)=kdep(i)+1
          go to 255
c
c ---- IF DATA TYPE IS TRACKLINE, BRANCH TO 600 ----
420       if(jc.eq.38) go to 600
c ---- IF DATA TYPE NOT A STATION, BRANCH TO 444 ----
          if(jc.ne.39) go to 444
c
c
c********* STATION DATA *********
          if(istn.eq.1) go to 424
          if(ista.ne.1) go to 428
c ---- STORE 'START' TIME FOR STATION IN 'tms1' ----
          tms1=1440*dy+60*hr+rmn
          istn=1
c ---- lst = true => THE START OF A STATION INTERVAL WHICH => ZERO SHIP VELOCITY
          lst=.true.
c ---- ksta COUNTS NUMBER OF STATIONS ----
          ksta=ksta+1
c ---- stan STORES UNIQUE STATION NUMBERS ENCOUNTERED; WHEN STATION RECORD
```

```
c        DOESN'T EXIST, THE STATION NUMBER WILL BE DETERMINED BY 'sp' VARIABLE
c        ON SAMPLE RECORDS ----
          stan(ksta)=isn
          tsst=tms1
          if(.not.lnv) go to 60
          lnu=.true.
          go to 850
424       lst=.false.
c ---- CHECKS CORRECT 'START/END' SEQUENCE FOR STATION ----
          if(ista.ne.2) go to 436
          tms2=1440*dy+60*hr+rmn
c ---- CALCULATES TIME DIFFERENCE BETWEEN 'ST/END' RECORDS & ADDS TO CUMULATIVE
c        TIME VARIABLE 'custa' ----
          custa=custa+(tms2-tms1)
          istn=2
          if(.not.lnv) go to 60
          lnu=.true.
          test=tms2
          go to 850
428       write(6,432) s(3),isn,dy,hr,rmn
432       format("ORECORD FOR STATION ",a1,a4," AT DAY/TIME ",i3,1x,i2,f4.1,
          " HAS INCORRECT ST/ED SEQUENCE."/"   TIME ON THIS STATION NOT AD",%
          "DED TO TOTAL.")
          istn=ista
c ---- istc COUNTS NUMBER OF STATION RECORDS OUT OF SEQUENCE ----
          istc=istc+1
          go to 60
436       write(6,440) s(3),isn,dy,hr,rmn
440       format("ORECORDS FOR STATION PREVIOUS TO ",a1,a4," WITH DAY/TIME",%
          " OF ",i3,"/",i2,f4.1," OUT OF SEQUENCE."/"   PREVIOUS STATIONS ",%
          "TIME NOT ADDED TO TOTAL.")
c ---- DROPS COUNT OF STATION FROM stan ARRAY BY 1 ----
          ksta=ksta-1
          istn=ista
          istc=istc+1
          go to 60
c ---- SENDS NAVIGATION DATA TYPE RECORDS TO STATEMENT 520 ----
444       if(hir.gt.104500) go to 520
c ---- SENDS PERSONNEL, PORT AND CRUISE DATA TYPE RECORDS TO STATEMENT 448 ----
          if(hir.lt.104000) go to 448
c
c
c********* EQUIPMENT, PORT/CRUISE, PERSONNEL DATA *********
          keq=keq+1
c ---- STORES EQUIPMENT TYPES IN 'iquip' ----
          iquip(keq)=jc
          hirq(keq)=s(20)
          go to 60
c ---- SENDS PERSONNEL DATA TYPE RECORDS TO STATEMENT 460 ----
448       if(jm.eq.2) go to 460
c ---- CONVERTS 'ST/ED' OF PORT TO 'ED/ST' SO CAN HAVE 'ST/ED/ST/ED' FOR ADDING
c        UP PORT TIMES ----
          kc=kc+1
          istb=ista
          if(jc.eq.2) istb=3-ista
          jdc(kc)=dy
          ihc(kc)=hr
          rmc(kc)=rmn
c ---- STORES CRUZ/PORT INFO FOR CURRENT RECORD IN 'idc' ----
          do 452 k=1,6
```

```
452       idc(kc,k)=idd(k)
          if(istb.eq.2) go to 456
          tmc1(kc)=1440*dy+60*hr+rmn
c ---- lstp = t => START OF PORT-STOP; tsst & test BORROWED TO RECORD TIME
         INTERVAL OF PORT STOP TO BE USED IN NAV-GAP OVERLAP ----
          lstp=.false.
          test=tmc1(kc)
          if(kc.gt.1) cuprt=cuprt+(tmc1(kc)-tmc2(kc-1))
          go to 60
456       tmc2(kc)=1440*dy+60*hr+rmn
          lstp=.true.
          tsst=tmc2(kc)
c ---- ADDS UP TOTAL TIME AT SEA FOR THE CRUISE ----
          cucrz=cucrz+(tmc2(kc)-tmc1(kc-1))
          go to 60
c ---- COUNTS PERSONNEL ----
460       if(kp.eq.0) go to 480
          kk=kp
c
c ---- COMPARES PERSONNEL NAME ON CURRENT RECORD TO THOSE PREVIOUSLY FOUND AND
c         STORED IN 'idp' ----
          do 470 k=1,kk
          if(idp(k,1).ne.idd(1)) go to 470
          if(idp(k,2).ne.idd(2)) go to 470
          if(idp(k,3).ne.idd(3)) go to 470
          go to 500
470       continue
480       kp=kp+1
c ---- STORES PERSONNEL NAME FOR CURRENT RECORD IN 'idp' ----
          do 490 j=1,6
490       idp(kp,j)=idd(j)
          hirp(kp)=s(20)
          k=kp
c ---- COUNTS ON-BOARD TIMES FOR CRUISE MEMBER.  SHOULD=2; IF NOT,
c         MUST DO INTERACTIVE FIXUP IN PRINT SUBROUTINE. ----
500       kper(k)=kper(k)+1
          if(kper(k).ne.1) go to 505
c ---- STORES INSTITUTIONAL AFFILIATION OF PERSONNEL ----
          paf(k)=isn
c ----STORES DUTY ASSIGNMENT OF PERSONNEL, I.E., WATCHSTANDER ----
          pjc(k)=jc
505       if(ista.eq.4) go to 510
c ---- STORES ON-BOARD TIMES FOR PERSONNEL IN 'jdp,ihp,rmp' ----
          if(ista.eq.0) go to 60
          jdp(k,1)=dy
          ihp(k,1)=hr
          rmp(k,1)=rmn
          go to 60
c ---- STORES OFF-BOARD TIMES FOR PERSONNEL ----
510       jdp(k,2)=dy
          ihp(k,2)=hr
          rmp(k,2)=rmn
          go to 60
c
c
c********* NAVIGATION DATA *********
520       if(kn.eq.0) go to 550
          nn=kn
c
c ---- TRIES TO MATCH CURRENT RECORD'S DATA TYPE TO DATA TYPES PREVIOUSLY STORED
```

```
c         IN 'INAV' ----
          do 540 i=1,nn
          if(jc.ne.inav(i)) go to 540
          kk=jmn(i)
c ---- TRIES TO MATCH CURRENT RECORD'S MEDIA TYPE TO MEDIA TYPES PREVIOUSLY
c         STORED IN 'KMD' ----
          do 530 k=1,kk
          if(jm.eq.kmd(i,k)) go to 570
530       continue
c ---- IF RECORD IS 'SYS' RECORD, RE-READ NEW RECORD ----
          if(jm.eq.7) go to 60
          go to 560
540       continue
550       if(jm.eq.7) go to 60
c ---- COUNTS UNIQUE DATA TYPES ENCOUNTERED ----
          kn=kn+1
c ---- STORES UNIQUE DATA TYPES ENCOUNTERED IN 'inav' ----
          inav(kn)=jc
          i=kn
c ---- COUNTS NUMBER OF UNIQUE MEDIA/DATA TYPE COMBINATIONS ----
560       jmn(i)=jmn(i)+1
          l=jmn(i)
c ---- STORES UNIQUE MEDIA TYPES ENCOUNTERED IN 'kmd' ----
          kmd(i,l)=jm
          go to 580
570       l=k
c ---- COUNTS # OF UNIQUE REELS, ROLLS, LISTS, ETC., FOR EACH DATA TYPE ----
580       if(ista.eq.1) knm(i,l)=knm(i,l)+1
          if(ista.eq.0) knm(i,l)=knm(i,l)+1
          if(.not.lnu) go to 60
          lnu=.true.
          go to 850
c
c
c********* CUMULATIVE DISTANCE ALGORITHM FOR ANALOG UNDERWAY DATA *********
c ---- lsoc = t => A SLAVE RECORD ----
600       if(s(3).ne.l1(5)) lsoc=.true.
          if(n.eq.0) go to 660
          nn=n
c
c ---- TRIES TO MATCH CURRENT RECORD'S DATA TYPE TO DATA TYPES PREVIOUSLY STORED
c         IN 'IDAT' ----
          do 640 i=1,nn
          if(jc.ne.idat(i)) go to 640
          kk=jmk(i)
c ---- TRIES TO MATCH CURRENT RECORD'S MEDIA TYPE TO MEDIA TYPES PREVIOUSLY
c         STORED IN 'imed' ----
          do 620 k=1,kk
          if(jm.ne.imed(i,k)) go to 620
          l=k
          if(lsoc) go to 800
          if(.not.lfg(i,k)) go to 700
          go to 682
620       continue
c ---- 'SYS' RECORDS TO 50.  READ NEW CDI RECORD ----
          if(jm.eq.7) go to 60
          go to 680
640       continue
c ---- n IS COUNT OF UNIQUE DATA TYPES ENCOUNTERED ----
660       if(jm.eq.7) go to 60
```

```
          n=n+1
          hirx(n)=s(20)
c ---- STORES UNIQUE DATA TYPES ENCOUNTERED IN 'idat' ----
          idat(n)=jc
          i=n
c ---- COUNTS THE # OF UNIQUE MEDIA TYPES FOR EACH DATA TYPE ----
680       jmk(i)=jmk(i)+1
          l=jmk(i)
c ---- STORES UNIQUE MEDIA TYPES ENCOUNTERED IN 'imed' ----
          imed(i,l)=jm
c ---- lfg = t INDICATES SEARCHING FOR PROPER 'ST/ON' SEQUENCE FOR 1ST RECORD
c        OF A PARTICULAR MEDIA/DATA TYPE COMBINATION ----
          lfg(i,l)=.true.
          sysdx(i,l)=0.0
          if(lsoc) go to 800
c ---- BRANCHES TO 798 IF 1ST RECORD FOR PARTICULAR DATA/MEDIA TYPE COMBINATION
c         IS A 'START/ON' ----
682       if(ista.eq.1.or.ista.eq.3) go to 798
c ---- COUNT OF DISRUPTIONS TO PROPER 'START/END/ON/OFF' SEQUENCE OF EACH
c        MEDIA/DATA SEQUENCE TYPE COMBINATION ----
          kerr(i,l)=kerr(i,l)+1
          write(6,685) ordnl(kerr(i,l)),s(1),s(2)
685       format("0",a8," RECORD OF DATA TYPE -",a3,1x,a4,"- IS NOT A STAR",%
          "T/ON."/"  TIME/DX ACCUMULATION WILL NOT BEGIN UNTIL A START/ON ",%
          "RECORD IS ENCOUNTERED.")
          write(6,318) ordnl(kerr(i,l)),dy,hr,rmn
          go to 60
c ---- BRANCHES, DEPENDING ON WHETHER PREVIOUS RECORD WAS 'ST/END/ON/OFF' ----
700       ii=ise(i,l)
          go to (720,760,720,780), ii
c ---- SENDS 'END/OFF' RECORDS TO STATEMENT 820 ----
720       if(ista.eq.2.or.ista.eq.4) go to 820
          kerr(i,l)=kerr(i,l)+1
          if(ista.ne.0) go to 730
723       write(6,726) s(1),s(2),dy,hr,rmn
726       format("0RECORD OF DATA TYPE -",a3,1x,a4,"- AND DAY/TIME ",i3,"/",
          i2,f4.1," HAS NO START/END/ON/OFF."/" A NEW RECORD WILL BE READ.")
          ise(i,l)=2
          go to 60
730       write(6,330) s(1),s(2),dy,hr,rmn
          write(6,740)
740       format("     IT WILL BE USED AS NEW STARTING POINT FOR ACCUMULATI",%
          "ING TIME/DX.")
          go to 800
c ---- SENDS 'START' RECORDS TO STATEMENT 800 ----
760       if(ista.eq.1) go to 800
          kerr(i,l)=kerr(i,l)+1
          if(ista.eq.0) go to 723
          write(6,330) s(1),s(2),dy,hr,rmn
          write(6,763)
763       format(" IF RECORD IS AN -ON-, IT WILL BE USED AS STARTING POINT",%
          " FOR ACCUMULATING TIME/DX;"/"  OTHERWISE, A NEW RECORD WILL BE ",%
          "READ.")
          if(ista.eq.3) go to 806
          go to 60
c ---- SENDS 'ON' RECORDS TO STATEMENT 806 ----
780       if(ista.eq.3) go to 806
          kerr(i,l)= kerr(i,l)+1
          if(ista.eq.0) go to 723
          write(6,330) s(1),s(2),dy,hr,rmn
```

```
           write(6,783)
783        format(" IF RECORD IS A -START-, IT WILL BE USED AS STARTING POI",%
           "NT FOR ACCUMULATING TIME/DX;"/"  OTHERWISE, A NEW RECORD WILL B",%
           "E READ.")
           if(ista.eq.1) go to 800
           go to 60
c ---- lfg = f => A 'START' RECORD OF PARTICULAR MEDIA/DATA TYPE COMBINATION
c         HAS BEEN ENCOUNTERED ----
798        lfg(i,l)=.false.
c ---- medk COUNTS # OF UNIQUE ROLLS, REELS, LISTS, ETC. FOR EACH DATA TYPE ----
800        if(ista.eq.1) medk(i,l)=medk(i,l)+1
           if(lsoc) go to 60
c ---- lso = t INDICATES CURRENT RECORD IS A 'START' OR AN 'ON' ----
806        lso=.true.
           tmg1(i,l)=1440*dy+60*hr+rmn
           go to 840
820        lso=.false.
           tmg2=1440*dy+60*hr+rmn
c ---- ADDS UP TOTAL RECORDING TIME FOR EACH MEDIA/DATA TYPE COMBINATION ----
           cugeo(i,l)=cugeo(i,l)+(tmg2-tmg1(i,l))
c ---- STORES WHETHER CURRENT RECORD IS A 'START/END/ON/OFF'; TO BE COMPARED
c         AGAINST NEXT RECORD OF SAME MEDIA/DATA TYPE ----
840        ise(i,l)=ista
c ---- IF NO NAVIGATION, ONLY CUMULATIVE RECORDING TIME, NOT DISTANCE, IS
c         DETERMINED ----
           if(.not.lnv) go to 60
c ---- IF PREVIOUS RECORD W/IN NAV. TIME-SPAN, BRANCHES TO 940 ----
850        if(lps) go to 940
           tdif1=1440*(dy-nvd1)+60*(hr-nvh1)+rmn-rmnv1
c ---- BRANCHES ACCORDING TO WHETHER TIME DIFFERENCE BETWEEN CURRENT CDI RECORD
c         AND 1ST NAV. RECORD IS <, =, > THAN ZERO ----
           if(tdif1) 860,900,880
860        if(.not.lnu) go to 870
           lnu=.false.
           go to 60
c ---- tbnv1 STORES TIME ACCUMULATED FOR EACH MEDIA/DATA TYPE BEFORE FIRST NAV
c         RECORD ENCOUNTERED ----
870        if(.not.lso) tbnv1(i,l)=cugeo(i,l)
           go to 60
c ---- IF CDI TIME > OR = TO 1ST NAV. RECORD TIME, lps = t AND WILL REMAIN
c         TRUE ----
880        lps=.true.
c ---- tlnv STORES DAY/TIME OF LAST RECORD TO HAVE NAVIGATION ----
           tlnv=1440*nvd1+60*nvh1+rmnv1
c ---- dlnv STORES CUMULATIVE DX AT DAY/TIME OF LAST RECORD TO HAVE NAVIGATION ----
           dlnv=cumdx
c ---- lnu = false => RECORD IS NOT ANALOG UNDERWAY DATA; BRANCH TO 940 ----
           if(lnu) go to 940
           if(lso) go to 940
           tbnv1(i,l)=cugeo(i,l)-(tmg2-tnv1)
c ---- SETS A DISTANCE MARK FROM CUMULATIVE DX ARRAY FOR CURRENT RECORD ----
           dxmrk(i,l)=cumdx
           go to 940
900        lps=.true.
           tlnv=1440*dy+60*hr+rmn
           dlnv=cumdx
           if(.not.lnu) go to 905
           lnu=.false.
           go to 60
905        if(.not.lso) go to 910
```

```
          dxmrk(i,l)=cumdx
          go to 60
910       tbnv1(i,l)=cugeo(i,l)
          go to 60
c ---- IF INDIVIDUAL RECORD IS WITHIN A NAVIGATION GAP, BRANCH TO 1000 ----
940       if(ilt.eq.0.and.iln.eq.0) go to 1000
          if(.not.lgap) go to 970
c ---- lend = true => RECORD IS THE FIRST TO HAVE NAVIGATION FOLLOWING A
c        NAVIGATION GAP ----
          lend=.true.
c ---- lgap = true => CURRENT RECORD IS IN A NAVIGATION GAP ----
          lgap=.false.
c ---- lpgp = true => PREVIOUS RECORD WAS IN A NAVIGATION GAP ----
          lpgp=.true.
          go to 1210
970       lpgp=.false.
          if(lnu) go to 1210
          if(lso) go to 1210
985       if(lend) lend=.false.
          delt=0
          npg=ng
          go to 1060
c ---- IF PREVIOUS RECORD WAS W/OUT NAVIGATION , BRANCH TO 1020 ----
1000      if(lgap) go to 1020
1005      lgap=.true.
          lpgp=.false.
c ---- ng COUNTS ALL THE NAVIGATION GAPS ENCOUNTERED ----
          ng=ng+1
c ---- tln/tfn ARRAYS STORE THE START/END TIMES FOR ALL NAVIGATION GAPS ENCOUNTERED.
c        THE START/END TIMES CONSIST OF THE TIMES FOR THE LAST AND FIRST RECORDS
c        WITH NAVIGATION WHICH BRACKET THE GAP ----
          tln(ng)=tlnv
c ---- dln/dfn STORE THE CUMULATIVE DISTANCE VALUES AT THE START/END TIMES OF A
c        NAVIGATION GAP ----
          dln(ng)=dlnv
c ---- IF RECORD NOT IN STATION INTERVAL, BRANCH TO 1010 ----
          if(lst) go to 1008
          if(lstp) go to 1008
          go to 1010
c ---- ksg IS COUNT OF STATION INTERVAL/NAVIGATION-GAP OVERLAP ----
1008      ksg=ksg+1
c ---- lap = true => RECORDS ARE WITHIN STATION/NAV-GAP OVERLAP ----
          if(.not.lsam) lap=.true.
c ---- tss STORES START OF STATION/NAVIGATION-GAP OVERLAP ----
          tss(ksg)=tlnv
          go to 1040
1010      if(test.le.tlnv) go to 1040
          ksg=ksg+1
          tss(ksg)=tlnv
          if(tsst.gt.tlnv) tss(ksg)=tsst
c ---- tes STORES END OF STATION/NAVIGATION-GAP OVERLAP ----
          tes(ksg)=test
          go to 1040
1020      lpgp=.true.
          if(lst.or.lstp) go to 1030
c ---- THE NEXT 8 STATEMENTS DETERMINE 'tss' AND 'tes' WHEN STATION WITHIN
c        NAV-GAP IS DETERMINED STRICTLY BY SAMPLE RECORDS, NOT STATION RECORDS
          if(.not.lsam) go to 1027
          if(.not.lfst) go to 1024
          ksg=ksg+1
```

```
      tss(ksg)=tsst
      if(tsst.le.tlnv) tss(ksg)=tlnv
      go to 1040
1024  tss(ksg)=tsst
      tes(ksg)=test
      go to 1040
1027  if(.not.lap) go to 1040
      tes(ksg)=test
      lap=.false.
      go to 1040
1030  if(lap) go to 1040
      ksg=ksg+1
      lap=.true.
      tss(ksg)=tsst
1040  if(.not.lbadnv) go to 1050
      ldis=.false.
      if(lnu.or.lso) go to 1243
      delt=tmg2-tlnv
      cuvt(i,l)=cuvt(i,l)+delt
      go to 1243
1050  if(lnu) go to 1210
      if(lso) go to 1210
      if(tmg1(i,l).gt.tlnv) go to 1120
c ---- delt STORES THE TIME BETWEEN SUCCESSIVE RECORDS WHICH DO NOT HAVE
c        NAVIGATION; THIS NO-NAV TIME IS ACCUMULATED FOR ALL UNDERWAY DATA
c        SYSTEMS IN 'cuvt' ----
      delt=tmg2-tlnv
      npg=ng-1
1060  if(npg.eq.0) go to 1130
      ig=npg
      do 1100 ip=1,npg
      if(tmg1(i,l).lt.tln(ig)) go to 1080
      if(tmg1(i,l).gt.tfn(ig)) go to 1130
      delt=tfn(ig)-tmg1(i,l)+delt
      go to 1130
1080  delt=tfn(ig)-tln(ig)+delt
1100  ig=ig-1
      go to 1130
1120  delt=tmg2-tmg1(i,l)
1130  if(ks.eq.0) go to 1200
      ks=ksg
c ---- NEXT 13 STATEMENTS SUBTRACT STATION/NAVIGATION-GAP OVERLAP TIMES OUT
c        OF 'delt', SINCE ON-STATION => ZERO SHIP VELOCITY ----
      do 1180 ip=1,ksg
      if(ks.lt.ksg) go to 1160
      if(.not.lap) go to 1160
      if(tss(ks).gt.tmg1(i,l)) go to 1140
      delt=0
      go to 1210
1140  delt=delt-tmg2+tss(ks)
      go to 1180
1160  if(tss(ks).gt.tmg1(i,l)) go to 1170
      delt=delt-tes(ks)+tmg1(i,l)
      go to 1200
1170  delt=delt-tes(ks)+tss(ks)
1180  ks=ks-1
c ---- 'cuvt' STORES TOTAL NO-NAVIGATION TIME FOR EACH DATA SYSTEM BUT DOES NOT
c        INCLUDE TIMES OF STATION/NAV-GAP OVERLAP ----
1200  cuvt(i,l)=cuvt(i,l)+delt
c ---- CALCULATES DIFFERENCE BETWEEN TIMES OF CURRENT CDI RECORD AND LAST
```

```
c        NAVIGATION RECORD READ ----
1210      tdif2=1440*(dy-nvd2)+60*(hr-nvh2)+rmn-rmnv2
c ---- IF CURRENT CDI RECORD TIME > CURRENT NAV. RECORD TIME, GO TO 1360 ----
          if(tdif2.gt.0) go to 1360
c ---- ldis = true => SUCCESSIVE RECORDS HAVE NAVIGATION, THEREFORE, DISTANCE
c        BETWEEN THEM CAN BE CALCULATED DIRECTLY ----
          ldis=.true.
          if(lgap.or.lpgp) ldis=.false.
1240      rlt=ilt/100000.
          rln=iln/100000.
c ---- CALLS 'DISTANCE' SUBROUTINE FOR CDI RECORDS OR THE COMBINATION OF A CDI
c        AND A NAVIGATION RECORD ----
1243      call dist (rlt,rln,dy,hr,rmn,vav)
          if(.not.lbadnv) go to 1250
          if(.not.lps) go to 1250
          write(6,1245) s(1),s(2),dy,hr,rmn
1245      format(" BAD NAV FOR RECORD ",a3,1x,a4," OF DAY/TIME ",i3,"/",i2,
          f4.1," TREATED AS IF NAV=0")
          go to 1005
c ---- TOTAL CUMULATIVE DX IS INCREMENTED ----
1250      cumdx=cumdx+dx
          if(lgap) go to 1270
          tlnv=1440*dy+60*hr+rmn
          dlnv=cumdx
          if(.not.lend) go to 1270
          if(.not.lap) go to 1260
c ---- 'tes' STORES TIME MARKING END OF SATION/NAV-GAP OVERLAP ----
          tes(ksg)=tlnv
          lap=.false.
c ---- 'tfn' STORES TIME MARKING END OF NAVIGATION GAP ----
1260      tfn(ng)=tlnv
          if(lnu) go to 1270
          if(.not.lso) go to 985
c ---- SHIFT CURRENT CDI RECORD VALUES INTO 1ST NAV. RECORD VARIABLES ----
1270      nvd1=dy
          nvh1=hr
          rmnv1=rmn
          rltn1=rlt
          rlnn1=rln
          if(.not.lnu) go to 1280
          lnu=.false.
          go to 1330
c ---- IF CURRENT CDI RECORD IS A 'ST/ON', ITS BENCHMARK IS SET IN TOTAL
c        CUMULATIVE DX ARRAY; OTHERWISE, BRANCH TO 1300 ----
1280      if(.not.lso) go to 1300
c ---- CUMULATIVE DX BENCHMARK IS SET FOR THE CURRENT PARTICULAR CDI MEDIA/DATA
c        TYPE RECORD ----
          dxmrk(i,l)=cumdx
          go to 1330
c ---- IF CURRENT CDI RECORD IS AN 'END/OFF', DIFFERENCE BETWEEN ITS LAST
c        CUMULATIVE DX BENCHMARK AND THE CURRENT TOTAL CUMULATIVE DX VALUE, IS
c        ADDED TO RECORD'S TOTAL CUMULATIVE DX ----
1300      sysdx(i,l)=sysdx(i,l)+cumdx-dxmrk(i,l)
          if(i.ne.1) go to 1330
1330      if(tdif2.ne.0) go to 60
          if(lmgvx) go to 48
          if(lmarc) go to 56
          go to 42
1360      ldis=.true.
          if(lpgp) ldis=.false.
```

```
c ---- CALL 'DISTANCE' SUBROUTINE FOR SUCCESSIVE NAVIGATION RECORDS ----
            call dist (rltn2,rlnn2,nvd2,nvh2,rmnv2,vav)
            if(.not.lbadnv) go to 1370
            go to 1410
c ---- ONLY TOTAL CUMULATIVE DX IS INCREMENTED BECAUSE NAV. RECORDS ARE WITHIN
c           'ST/ON - END/OFF' SEQUENCES FOR ALL CURRENT CDI RECORDS ----
  1370      lpgp=.false.
            cumdx=cumdx+dx
            tlnv=1440*nvd2+60*nvh2+rmnv2
            dlnv=cumdx
            if(.not.lend) go to 1390
            if(.not.lap) go to 1380
            tes(ksg)=tlnv
            lap=.false.
  1380      tfn(ng)=tlnv
            dfn(ng)=dlnv
            lend=.false.
c ---- SHIFT 2ND NAV. RECORD VALUES INTO 1ST NAV. RECORD VARIABLES -----
  1390      nvd1=nvd2
            nvh1=nvh2
            rmnv1=rmnv2
            rltn1=rltn2
            rlnn1=rlnn2
c ---- READ NEW VALUES INTO 2ND NAV. RECORD VARIABLES ----
  1410      if(lmgvx) go to 1420
            if(lmarc) go to 1460
            read(10,44,end=60) nvd2,nvh2,rmnv2,rltn2,rlnn2
            go to 1210
  1420      read(10,end=60) nvd2,nvh2,rmnv2,rltn2,rlnn2
            go to 1210
  1460      read(10,end=60) nvd2,nvh2,dam2,iline,ishot,itype,dlat2,dsdlat,
            dlon2,dsdlon,dcse,dspd,dumya8
            rmnv2=sngl(dam2)
            rltn2=sngl(dlat2)
            rlnn2=sngl(dlon2)
            go to 1210
  1500      call print
  1600      endfile 42
            call io ("close","file08")
            call io ("detach","file08")
            call io ("close","file42")
            call io ("detach","file42")
            stop
            end
```

```
COMPILATION LISTING OF SEGMENT print
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1656.1 pst Tue
    Options: table symbols
```

```
c********* PRINT SUBROUTINE *********
           subroutine print
c
           logical lgel,lgep,lhyd,lnv,lsad,lstat
c
           character*1 let,shp,un
           character*2 b2,cloc,lett,sh,shpp
           character*4 area,aria(5)
           character*8 chief,depl,dout,dsh,ibk,loc,mout,quip,ship,shpname,
                       und
c
           integer are,dsam,dyl,hirg,hirh,hirp,hirq,hirx,hrl,ib,pjc,s
c
           real nauts
c
           dimension chief(4),dyl(30),hrl(30),iqp(60),mon(30),ndl(10)
           ,ni(70),np(20),nt(20),quip(120),shpname(2)
c
           common/stprdt/idc(30,6),jdc(30),ihc(30),kc,s(20)
           common/blok/dout(316),dsh,loc(5),mout(30),ship(26),und,
           are(10),area(50),type(12),month(12),shp(10),shpp(3),b2
           common/stpr/ibk,cucrz,cugeo(30,10),cuhyd(30,10),cuprt,custa,dsam(3%
           0,3),paf(70),rmc(30),rmp(70,2),rnot,sysdx(30,10),hirg(30),hirh(30)
           ,hirp(70),hirq(60),hirx(30),ib,idat(30),ide(10),idp(70,6),ihyd(20),
           ihir(10),ihp(70,2),imed(30,10),inav(30),iob(10),iquip(60),isam(30),
           itrk,un,jdp(70,2),jmk(30),jml(10),jmn(10),kdep(10),keq,khym(30,10),
           kmd(30,10),kmed(30,10),kn,knm(30,10),kob(10),kp,kper(70),ksam,
           ksn(10),ksta,medk(30,10),n,nd,nh,no,nrec(30),nsam(30),pjc(70),lgep,
           lgel,lhyd,lnv,lsad,lstat
c
           data nlpo,nsa,nsr,ns1,ns2,ns3/6*0/
           call blockdata
           if(lnv) go to 20
c
c
c********* PRINTS CRUISE REPORT HEADER *********
           write(42,15) und,und,und
 15        format(" ",27x,"PRELIMINARY CRUISE REPORT"/"+",27x,3a8,"_"/" ",29x
           ,"(NAVIGATION PENDING)")
           nlpo=nlpo+1
           go to 45
 20        write(42,30) und
 30        format(" ",33x,"CRUISE REPORT"/"+",33x,a8,"_____")
 45        write(42,60)
 60        format("0",36x,"OF THE"/" ",27x,"BRANCH OF MARINE GEOLOGY")
           write(42,90)
 90        format(" ",20x,"U.S. GEOLOGICAL SURVEY, MENLO PARK, CA")
c ---- cloc STORES RIGHT-MOST 2 LETTER/NUMBERS OUT OF CRUISE LOCATOR 's(13)' ----
           decode(s(13),100) cloc
 100       format(2x,a2)
c ---- sh STORES LEFT-MOST 2 LETTERS OUT OF (s(13)' ----
           decode(s(13),110) sh
 110       format(a2)
c ---- PICKS APPROPRIATE FORMAT FOR xxxx TYPE CRUISE LOCATOR (EG. sea3) ----
           if(sh.ne.b2) write(42,122) s(13),s(14),s(15)
```

```
c ---- PICKS APPROPRIATE FORMAT FOR __xx TYPE OF CRUISE LOCATOR (EG. __s3) ----
        if(sh.eq.b2) write(42,120) cloc,s(14),s(15)
120     format("0",38x,"FOR"/"0",31x,"CRUISE -",a2,"-",i2,"-",a2,"-")
122     format("0",38x,"FOR"/"0",29x,"CRUISE -",a4,"-",i2,"-",a2,"-")
        write(42,150)
150     format(//)
c
c
c********* PRINTS GENERAL CRUISE INFORMATION *********
        write(42,180) dsh,dsh,dsh,dsh,dsh,dsh
180     format(" ",3a8,"--"/" GENERAL CRUISE INFORMATION"/" ",3a8,"--")
c
c ---- CRUISE AREA ----
        write(6,210)
210     format("0GIVE A MORE EXACT LOCATION OF THIS CRUISE.  USE 40 CHAR",%
        "ACTERS OR LESS."/"   E.G., POINT MENDECINO CAL. TO COOS BAY ORE",%
        ".; MONTEREY-DELGADO FAN; BEAUFORT SEA;"/"   COOK INLET TO KODIA",%
        "K ISLAND; NORTON SOUND, NORTHERN BERING; ETC.")
        read(5,240) loc
240     format(5a8)
        do 270 i=1,10
        if(s(15).eq.are(i)) go to 330
270     continue
        write(6,310)
310     format(" ENTER AREA. (5a4)")
        read(5,315) aria
315     format(5a4)
        write(42,360) (aria(j),j=1,5), loc
        go to 374
330     ii=5*i-4
        jj=ii+4
        write(42,360) (area(j),j=ii,jj),loc
360     format("0",3x,"AREA:  ",5a4,"  /  ",5a8/"+",3x,"____")
        decode(s(13),363) let
363     format(2x,a1)
        decode(s(13),366) lett
366     format(1x,a2)
c
c ---- SEARCHES FOR SHIPNAME IN DATA STATEMENT 'shp' AND 'ship' ----
374     do 390 i=1,10
        if(let.ne.shp(i)) go to 390
        if(i.gt.3) go to 490
        do 380 j=1,3
        if(lett.ne.shpp(j)) go to 380
        go to 480
380     continue
        go to 490
390     continue
        write(6,420)
420     format("0ENTER THE SHIPS NAME. (2A8)")
        read(5,450) shpname
450     format(2a8)
        go to 510
480     i=j+10
490     ii=2*i-1
        jj=ii+1
        shpname(1)=ship(ii)
        shpname(2)=ship(jj)
510     write(42,540) shpname
540     format("0",3x,"SHIP:  R/V ",2a8/"+",3x,"____")
```

```
c
c ---- PRINTS OUT NAME OF CHIEF SCIENTIST ----
      write(6,570)
570   format("0ENTER THE NAME(S) OF THE CHIEF SCIENTIST(S) FOR THIS CR",%
      "UISE.  IF 2 INDIVIDUALS INVOLVED, SEPARATE"/"   THEIR NAMES WIT",%
      "H A /. (4a8)")
      read(5,240) chief
      write(42,600) chief, und,und
600   format("0",3x,"CHIEF SCIENTIST(S):  ",4a8/"+",3x,2a8,"__")
      ii=1
c
c ---- ASSESES WHETHER GEOPHYSICAL, GEOLOGICAL OR HYDROGRAPHIC DATA PRESENT ----
      if(.not.lgep) go to 630
      if(.not.lgel) go to 660
      ij=4
      if(.not.lhyd) go to 780
      ik=7
      go to 810
630   if(.not.lgel) go to 690
      ii=4
660   if(.not.lhyd) go to 750
      ij=7
      go to 780
690   ii=7
      go to 750
720   ii=4
750   ij=10
780   ik=10
810   ji=ii+2
      jj=ij+2
      jk=ik+2
      write(42,840) (type(i),i=ii,ji),(type(i),i=ij,jj),(type(i),i=ik,jk
      ),und
840   format("0",3x,"TYPE OF DATA"/" ",3x,"COLLECTED:     ",3a4,", ",3a4,
      ", ",3a4/"+",3x,a8,"_")
c
c ---- CONVERTS GMT AND JULIAN DAY OF START/END OF CRUISE AND PORT STOPS TO
c        LOCAL DATE AND TIME AND PRINTS THEM OUT ----
      call datcon (mon,dyl,hrl)
      write(42,900)
      write(42,870) und,und,und,und,und,und
870   format("0",3x,"CRUISE DATES: ",4x,"LOCAL DATE/TIME*",3x,"TIME(JD",%
      "/GMT)",13x,"PORT"/"+",3x,a8,"____",6x,a8,"_______",4x,a8,"____",
      3x,3a8)
      write(42,900)
900   format(" ")
c ---- PRINTS START/END OF CRUISE ----
      write(42,930)
930   format(" ",5x,"START CRUISE:")
      imc=ifix(rmc(1))
      write(42,960) dyl(1),month(mon(1)),hrl(1),imc,jdc(1),ihc(1),imc,
      (idc(1,j),j=1,6)
960   format("+",21x,i2,1x,a3,1x,2i2,1x,"HRS",6x,i3,"/",2i2,5x,6a4)
      write(42,900)
      imc=ifix(rmc(kc))
      write(42,960) dyl(kc),month(mon(kc)),hrl(kc),imc,jdc(kc),ihc(kc),
      imc,(idc(kc,j),j=1,6)
      write(42,990)
990   format("+",5x,"END   CRUISE:")
      nlpo=nlpo+30
```

```
          if(kc.lt.4) go to 1150
          write(42,900)
c ---- PRINTS PORT TIMES ----
          write(42,1020) und
1020      format("0",3x,"PORT STOPS:"/"+",3x,a8,"__")
          nlpo=nlpo+3
          ii=0
          m=1
          km=kc-2
c ---- DETERMINES WHETHER 'ARRIVE' OR 'LEAVE' SHOULD BE PRINTED ----
          do 1140 i=1,km
          m=m+1
          if(mod(i,2).eq.0) go to 1080
          ii=ii+1
          write(42,900)
          write(42,1050) ii
1050      format(" ",5x,i2,".",2x,"ARRIVE:")
          go to 1125
1080      write(42,1110)
1110      format(" ",11x,"LEAVE:")
          nlpo=nlpo+3
1125      imc=ifix(rmc(m))
1140      write(42,960) dyl(m),month(mon(m)),hrl(m),imc,jdc(m),ihc(m),imc,
          (idc(m,j),j=1,6)
1150      write(42,1170)
1170      format("0",3x,"* EXPRESSED IN LOCAL STANDARD TIME.")
          write(42,900)
          nlpo=nlpo+3
          if((nlpo+6).lt.61) go to 1193
          write(42,1185)
1185      format("1")
          nlpo=1
c
c ---- PRINTS TOTAL UNDERWAY TIME AND TOTAL PORT TIME IN HOURS AND DAYS AND
c        HOURS ----
c ---- ud = DAYS OF UNDERWAY TIME ----
1193      ud=cucrz/1440.
c ---- iour = TOTAL UNDERWAY TIME IN HOURS ----
          iour=cucrz/60
          iud=ifix(ud)
c ---- iuhr IS HOURS OF UNDERWAY TIME BEYOND AN INTEGER NUMBER ----
          iuhr=(cucrz-1440*iud)/60
c ---- pd = DAYS OF PORT TIME ----
          pd=cuprt/1440.
c ---- jour = TOTAL PORT TIME IN HOURS ----
          jour=cuprt/60
          ipd=ifix(pd)
c ---- iphr = HOURS OF PORT TIME BEYOND INTEGER NUMBER OF DAYS OF PORT TIME ----
          iphr=(cuprt-1440*ipd)/60
          write(42,1200) und,und
1200      format("0",29x,"HOURS",10x,"DAYS & HOURS"/"+",29x,"_____",8x,2a8)
          write(42,1210) iour,iud,iuhr,und,und
1210      format("0",3x,"TOTAL UNDERWAY TIME:",6x,i4,9x,i3," DAYS  ",i2,
          " HRS"/"+",3x,2a8,"___")
          nlpo=nlpo+4
          if(kc.lt.3) go to 1233
          write(42,1230) jour,ipd,iphr,und
1230      format("0",3x,"TOTAL PORT TIME:",10x,i4,9x,i3," DAYS  ",i2,
          " HRS"/"+",3x,a8,"_______")
          nlpo=nlpo+2
```

```
1233       write(42,150)
           nlpo=nlpo+3
           if((nlpo+9).lt.61) go to 1235
           write(42,1185)
           nlpo=1
c
c
c********* PERSONNEL LIST *********
1235       write(42,1240) dsh,dsh
1240       format(" ",a8,"------"/" PERSONNEL LIST"/" ",a8,"------")
           if(kp.eq.0) go to 1375
c ---- SORT 'HIERARCHALLY' ORDERS PERSONNEL LIST ----
           call sort (kp,hirp,ni)
           write(42,1250) und,und,und,und,und,und,und
1250       format("0",13x,"NAME",12x,"AFFIL",7x,"DUTIES",8x,"ABOARD",4x,
           "ASHORE"/"+",3x,3a8,2x"_____",2x,2a8,2x,a8,2x,a8)
           write(42,900)
           nlpo=nlpo+6
           npr=0
c
c ---- SECTION ENABLES USER TO SELECT OUT ONE CORRECT SPELLING OF DUPLICATELY
c         SPELLED NAMES ----
c ---- LISTS AND NUMBERS DUPLICATE NAMES ----
           do 1280 i= 1,kp
           j=ni(i)
           if(kper(j).eq.2) go to 1280
           npr=npr+1
           np(npr)=j
           if(npr.ne.1) go to 1265
           write(6,1260)
1260       format("OLIST OF NAMES LOGGED ONLY ONCE FOLLOWS, CHECK SPELLING ",%
           "INCONSISTENCIES:")
           write(6,900)
1265       write(6,1270) npr,(idp(j,k),k=1,6),(jdp(j,k),k=1,2)
1270       format(" ",i2,1x,6a4,3x,"JD ",i3,"/",i3)
1280       continue
           write(6,1290)
1290       format("0??HOW MANY PAIRS OF DUPLICATE NAMES ARE THERE?? (I2 W/L",%
           "EADING ZEROS)")
           read(5,1295) npar
1295       format(i2)
           if(npar.eq.0) go to 1330
           write(6,1300)
1300       format("OFOR EACH PAIR OF DUPLICATE NAMES, ENTER THEIR NUMBERS F",%
           "ROM THE ABOVE LIST."/" ENTER THEM TWO AT A TIME LISTING THE NUM",%
           "BER TO BE DELETED, FIRST. (I2,1X,I2 FORMAT W/LEADING ZEROS). EN",%
           "TERING  03 04 MEANS NAME 3 WILL BE DELETED.  ENTRIES PROMPTED B",%
           "Y >>")
c
c ---- USER ENTERS NUMBERS AND DUPLICATE NAMES TO BE DELETED ----
           do 1320 i=1,npar
           write(6,1305)
1305       format("0>>")
           read(5,1310) ndl(i),nkp
1310       format(i2,1x,i2)
           j=np(ndl(i))
           jj=np(nkp)
           if(jdp(jj,1).ne.0) go to 1315
c ---- 'ON OR OFF' TIME OF NAME TO BE DELETED IS COMBINED WITH 'ON OR OFF'
c         TIME OF NAME TO BE KEPT ----
```

```
           jdp(jj,1)=jdp(j,1)
           ihp(jj,1)=ihp(j,1)
           rmp(jj,1)=rmp(j,1)
           go to 1320
1315       jdp(jj,2)=jdp(j,2)
           ihp(jj,2)=ihp(j,2)
           rmp(jj,2)=rmp(j,2)
1320       continue
c
c ---- DETERMINES AFFILIATION AND DUTIES OF EACH PERSON LISTED ----
1330       do 1360 i=1,kp
           j=ni(i)
           if(npar.eq.0) go to 1335
           do 1333 jl=1,npar
           kl=ndl(jl)
           if(np(kl).eq.j) go to 1360
1333       continue
1335       kk=2*pjc(j)-1
           irmp1=ifix(rmp(j,1))
           irmp2=ifix(rmp(j,2))
c
c ---- PRINTS OUT PERSONNEL NAMES, AFFILIATIONS, DUTIES AND TIMES ABOARD
c         ASHORE ----
           write(42,1338) (idp(j,k),k=1,6),paf(j),(dout(k),k=kk,kk+1)
1338       format(" ",6x,6a4,a4,2x,2a8)
           nlpo=nlpo+1
           if(jdp(j,1).eq.0) go to 1345
           write(42,1340) jdp(j,1),ihp(j,1),irmp1
1340       format("+",54x,i3,"/",2i2)
1345       if(jdp(j,2).eq.0) go to 1360
           write(42,1350) jdp(j,2),ihp(j,2),irmp2
1350       format("+",64x,i3,"/",2i2)
1360       continue
c
c
c********* EQUIPMENT SYSTEMS USED *********
           if(keq.eq.0) go to 1758
1375       in=0
           ip=0
           ig=0
           ih=0
           ni(1)=1
           if(keq.eq.1) go to 1377
           call sort (keq,hirq,ni)
c
c ---- CHECKS HIERARCHAL NO. OF EACH PIECE OF EQUIPMENT AND COUNTS THE NUMBER
c         OF EACH TYPE ----
1377       do 1500 i=1,keq
           j=ni(i)
           ihr=(hirq(j)-104000)/100+1
           go to (1380,1410,1440,1470),ihr
c ---- COUNTS NAVIGATIONAL EQUIPMENT ----
1380       in=in+1
           go to 1500
c ---- COUNTS GEOPHYSICAL EQUIPMENT ----
1410       ip=ip+1
           go to 1500
c ---- COUNTS GEOLOGIC EQUIPMENT ----
1440       ig=ig+1
           go to 1500
```

```
c ---- COUNTS HYDROGRAPHIC EQUIPMENT ----
1470      ih=ih+1
1500      continue
c ---- mx IS THE MAXIMUM OF in, ip, ig AND ih ----
          mx=max0(in,ip,ig,ih)
c ---- EXPANDS mx BY 4 TO = THE ELEMENTS IN A '4 BY mx' MATRIX ----
          mx=4*mx
c
c ---- INITIALIZE iqp, k, ix, iy, and nm ----
          do 1530 i=1,mx
1530      iqp(i)=0
          k=1
          ix=1
          iy=in
c
c ---- PUTS DATA CODE POSITION NUMBER 'jc' (STORED IN iquip) FOR A PARTICULAR
c           TYPE OF EQUIPMENT INTO EVERY FOURTH ELEMENT OF iqp STARTING AT nm(th)
c           ELEMENT ----
1560      nm=k
          if(ix.gt.iy) go to 1620
          do 1590 i=ix,iy
          j=ni(i)
          iqp(nm)=iquip(j)
1590      nm=nm+4
1620      k=k+1
          ix=iy+1
          if(k.eq.2) iy=iy+ip
          if(k.eq.3) iy=iy+ig
          if(k.eq.4) iy=iy+ih
          if(k.lt.5) go to 1560
          write(42,150)
          nlpo=nlpo+3
          nlpo=mod(nlpo,60)
          if((nlpo+9).lt.61) go to 1635
          write(42,1185)
          nlpo=1
1635      write(42,1650) dsh,dsh,dsh,dsh
1650      format(" ",2a8,"------"/" EQUIPMENT SYSTEMS USED"/" ",2a8,"------"
          )
          write(42,1680) und,und,und,und,und,und,und,und
1680      format("0",3x," NAVIGATIONAL  ",3x," GEOPHYSICAL   ",3x,"   GE",%
          "OLOGICAL   ",3x," HYDROGRAPHICAL "/"+",3x,2a8,3x,2a8,3x,2a8,3x,
          2a8/)
          nlpo=nlpo+6
c
c ---- DETERMINES FULL NAME OF EACH PIECE OF EQUIPMENT USED ----
          do 1710 i=1,mx
          l=2*i-1
          if(iqp(i).eq.0) go to 1700
          ii=2*iqp(i)-1
          ij=ii+1
c ---- dout = dot1, dot2, dot3 OF DATA STATEMENT IN SUBROUTINE BLOCKDATA ----
          quip(l)=dout(ii)
          quip(l+1)=dout(ij)
          go to 1710
1700      quip(l)=ibk
          quip(l+1)=ibk
1710      continue
          mx=mx/4
          l=1
```

```
c
c ---- PRINTS OUT EQUIPMENT LIST ----
c ---- dout = dot1, dot2, dot3 of DATA STATEMENT USEDIN SUBROUTINE BLOCKDATA ----
        do 1755 i=1,mx
        ll=l+7
        write(42,1740) (quip(j),j=l,ll)
1740    format(" ",3x,2a8,3x,2a8,3x,2a8,3x,2a8)
        nlpo=nlpo+1
        l=l+8
1755    continue
1758    write(42,150)
        nlpo=nlpo+3
        nlpo=mod(nlpo,60)
        if((nlpo+12).lt.61) go to 1760
        write(42,1185)
        nlpo=1
c
c
c********* PRINTS TRACKLINE DISTANCES, TOTAL RECORDING HOURS AND DATA QUANTI-
c           TIES FOR EACH UNDERWAY DATA TYPE *********
1760    write(42,1770) dsh,dsh
1770    format(" ",a8,"------"/" DATA COLLECTED"/" ",a8,"------")
        if(n.eq.0) go to 2130
        write(42,1800) und
1800    format("0",3x,"GEOPHYSICAL"/"+",3x,a8,"___")
        write(42,1830) und,und,und,und,und,und,und
1830    format("0",8x,"DATA TYPE",8x,"RECORDING",4x,"TRACKLINE",1x,"TRAC",%
        "KLINE",1x,"RECORDING",1x,"ROLL,REEL"/" ",8x,"OR SYSTEM",10x,
        "MEDIUM",5x,"KILOMETRS",1x,"N. MILES ",1x,"TIME(HRS)",1x,
        "LIST QTY"/"+",5x,2a8,2x,a8,"______",4(1x,a8,"_"))
        nlpo=nlpo+8
        ni(1)=1
        if(n.eq.1) go to 1840
c ---- HIERARCHALLY SORTS DATA TYPES ----
        call sort (n,hirx,ni)
c
c ---- CONVERTS MINUTES TO HOURS, DETERMINES SYSTEM AND MEDIUM
c        NAMES AND PRINTS IT ALL OUT ----
1840    do 1920 i=1,n
        j=ni(i)
c ---- DETERMINES FULL NAME OF EACH TYPE OF DATA -----
        k=2*idat(j)-1
        if(k.ne.75) go to 1845
        itrk=j
c ---- IF TRACKLINES ENCOUNTERED IN idat, GO TO 1920 ----
        go to 1920
1845    kk=k+1
c ---- PRINTS OUT DATA TYPE NAME ----
        write(42,1860) dout(k),dout(kk)
1860    format("0",5x,2a8)
        nlpo=nlpo+2
        ll=jmk(j)
        do 1920 l=1,ll
c ---- CONVERTS RECORDING MINUTES TO DECIMAL HOURS ----
        hours=cugeo(j,l)/60.
        if(lnv) go to 1870
c ---- CALCULATES NAUTICAL MILES AND KILOMETERS OF RECORDING DX USING AVERAGE
c        SHIP SPEED AND RECORDING HOURS ----
        nauts=rnot*hours
        sysdx(j,l)=nauts/.5399568
        go to 1880
c ---- CONVERTS KILOMETERS TO NAUTICAL MILES FOR CASE WHERE NAVIGATION EXISTS ----
1870    nauts=sysdx(j,l)*.5399568
c ---- DETERMINES FULL NAME OF EACH RECORDING MEDIUM ----
1880    k=2*imed(j,l)-1
        kk=k+1
        if(l.lt.2) go to 1884
        write(42,900)
        nlpo=nlpo+1
1884    write(42,1890) mout(k),mout(kk),sysdx(j,l),nauts,hours,medk(j,l)
1890    format("+",23x,a8,a6,1x,f7.1,3x,f7.1,3x,f7.1,6x,i2)
1920    continue
        if(lnv) go to 1970
        write(42,1940) rnot
1940    format("0",3x,"**NOTE:  NAVIGATION PENDING, THEREFORE, TRACKLINE",%
        " DISTANCES ARE BASED ON"/" ",12x,"RECORDING TIMES MULTIPLIED BY",%
        " AN ESTIMATED SHIP SPEED OF ",f4.1," KNOTS.")
        nlpo=nlpo+3
1970    write(42,900)
        nlpo=nlpo+1
        nlpo=mod(nlpo,60)
        if((nlpo+9).lt.61) go to 2130
        write(42,1185)
        nlpo=1
c
c
c********* PRINTS OUT GEOLOGIC/HYDROLOGIC SAMPLES BY TYPE, NUMBER, AND
c         DEPTH *********
2130    if(ksam.eq.0) go to 2280
        write(42,2160) und,und,und
2160    format("0",3x,"GEOLOGICAL/HYDROLOGICAL SAMPLES"/"+",3x,3a8,"____",%
        "___")
        write(42,2190) un,un,un,und,und,und,und,und,und,und
2190    format(" ",48x,"NUMBER OF SAMPLES FROM A"/" ",6x,"SAMPLING",10x,
        "SAMPLING ",1x,"SAMPLES  ",3x," GIVEN WATER DEPTH INTERVAL  "/" ",%
        6x,"DEVICE",12x,"ATTEMPTS ",1x,"RECOVERED",3x," 0-100",a1,3x,"10",%
        "0-3000",a1,3x,">3000",a1/"+",6x,2a8,2x,a8,"_",1x,a8,"_",2x,
        3(1x,a8,"_"))
        write(42,900)
        nlpo=nlpo+6
        ni(1)=1
        if(ksam.eq.1) go to 2205
c ---- SORTS SAMPLES HIERARCHALLY BY TYPE ----
        call sort (ksam,hirg,ni)
c
c ---- DETERMINES NUMBER OF SAMPLE ATTEMPTS, RECOVERIES, AND RECOVERIES BY
c      DEPTH ----
2205    do 2250 i=1,ksam
        j=ni(i)
c ---- DETERMINES NAME OF SAMPLING DEVICE ----
        k=2*isam(j)-1
        kk=k+1
c ---- nsat = NO. OF SAMPLE ATTEMPTS FOR EACH TYPE ----
        nsat=nrec(j)+nsam(j)
c ---- nsa = NO. OF TOTAL SAMPLE ATTEMPTS FOR EACH TYPE ----
        nsa=nsat+nsa
c ---- nsr = TOTAL SAMPLE RECOVERIES ----
        nsr=nsam(j)+nsr
c ---- ns1, ns2, ns3 = TOTAL SAMPLES RECOVERED FROM DEPTHS OF 0-100,
c      100-3000, AND >3000 METERS ----
```

```
          ns1=dsam(j,1)+ns1
          ns2=dsam(j,2)+ns2
          ns3=dsam(j,3)+ns3
          write(42,2220) dout(k),dout(kk),nsat,nsam(j),dsam(j,1),dsam(j,2),
          dsam(j,3)
2220      format(" ",6x,2a8,5x,i3,7x,i3,2x,3(7x,i3))
          nlpo=nlpo+1
2250      continue
          write(42,2260)
2260      format(" ",25x,"------",4x,"------",6x,"------",4x,"------",4x,
          "------")
          write(42,2270) nsa,nsr,ns1,ns2,ns3
2270      format(" ",6x,"TOTALS",14x,i4,6x,i4,8x,i4,6x,i4,6x,i4)
          nlpo=nlpo+2
2280      if(nh.eq.0) go to 2460
          write(42,900)
          nlpo=nlpo+1
          nlpo=mod(nlpo,60)
          if((nlpo+9).lt.61) go to 2295
          write(42,1185)
          nlpo=1
c
c
c********* PRINTS OUT GEOLOGIC/HYDROGRAPHIC ANALOG DATA QUANTITIES AND
c             RECORDING HOURS *********
2295      write(42,2310) und,und,und,und
2310      format("0",3x,"GEOLOGICAL/HYDROLOGICAL (ANALOG)"/"+",3x,4a8)
          write(42,2340) und,und,und,und,und,und
2340      format("0",9x,"DATA TYPE     ",6x,"RECORDING    ",5x,"RECORDING",4x,
          "NUMBER OF TAPES,"/" ",9x,"OR SYSTEM     ",8x,"MEDIUM     ",5x,"TI",%
          "ME(HRS)",4x,"ROLLS,LISTS,ETC."/"+",6x,2a8,4x,a8,"______",5x,a8,
          "_",4x2a8)
          nlpo=nlpo+5
          ni(1)=1
          if(nh.eq.1) go to 2355
c   ---- HIERARCHALLY SORTS DATA ----
          call sort (nh,hirh,ni)
c
c   ---- DETERMINES FULL NAME OF EACH DATA TYPE ----
2355      do 2430 i=1,nh
          j=ni(i)
          k=2*ihyd(j)-1
          kk=k+1
          write(42,2370) dout(k),dout(kk)
2370      format("0",6x,2a8)
          nlpo=nlpo+2
          ll=jml(j)
          do 2430 l=1,ll
c   ---- CONVERTS TOTAL RECORDING MINUTES TO HOURS ----
          hours=cuhyd(j,l)/60.
c   ---- DETERMINES FULL RECORDING MEDIA NAME FOR EACH DATA TYPE ----
          k=2*kmed(j,l)-1
          kk=k+1
          if(l.le.1) go to 2395
          write(42,900)
          nlpo=nlpo+1
2395      write(42,2400) mout(k),mout(kk),hours,khym(j,l)
2400      format("+",26x,a8,a6,7x,f5.1,12x,i3)
2430      continue
2460      if(no.eq.0) go to 2640
```

```
        write(42,900)
        nlpo=nlpo+1
        nlpo=mod(nlpo,60)
        if((nlpo+9).lt.61) go to 2470
        write(42,1185)
        nlpo=1
c
c
c********* PRINTS OUT NUMERICAL OBSERVATIONS *********
2470    write(42,2480) und,und
2480    format("0",3x,"NUMERICAL OBSERVATION"/"+",3x,2a8,"______")
        write(42,2490) und,und,und,und
2490    format("0",15x,"DATA TYPE",8x,"NUMBER OF",4x,"TAKEN OVER HOW"/" "
        ,15x,"OR SYSTEM",8x,"READINGS",5x,"MANY STATIONS"/"+",12x,2a8,4x,
        a8,"_",4x,a8,"______")
        write(42,900)
        nlpo=nlpo+6
        nt(1)=1
        if(no.eq.1) go to 2463
        call sort (no,ihir,nt)
2463    do 2610 i=1,no
        j=nt(i)
c ---- DETERMINES FULL NAME OF EACH DATA TYPE ----
2550    k=2*iob(j)-1
        kk=k+1
        write(42,2580) dout(k),dout(kk),kob(j),ksn(j)
2580    format(" ",12x,2a8,7x,i3,13x,i3)
        nlpo=nlpo+1
2610    continue
2640    if(kn.eq.0) go to 2820
        write(42,900)
        nlpo=nlpo+1
        nlpo=mod(nlpo,60)
        if((nlpo+10).lt.61) go to 2655
        write(42,1185)
        nlpo=0
c
c
c********* PRINTS INFORMATION ON NAVIGATION DATA *********
2655    write(42,2670) und
2670    format("0",3x,"NAVIGATIONAL"/"+",3x,a8,"____")
        write(42,2700) und,und,und,und,und
2700    format("0",15x,"DATA TYPE",10x,"RECORDING",7x,"NUMBER OF TAPES,"
        /" ",15x,"OR SYSTEM",12x,"MEDIUM",8x,"ROLLS,LISTS,ETC."/"+",12x,
        2a8,4x,a8,"______",4x,2a8)
        nlpo=nlpo+5
        do 2790 i=1,kn
c ---- DETERMINES FULL NAME OF EACH NAVIGATION SYSTEM ----
        k=2*inav(i)-1
        kk=k+1
        write(42,2730) dout(k),dout(kk)
2730    format("0",12x,2a8)
        nlpo=nlpo+2
        ll=jmn(i)
        do 2790 l=1,ll
c ---- DETERMINES RECORDING MEDIUM FOR EACH NAVIGATION SYSTEM ----
        k=2*kmd(i,l)-1
        kk=k+1
        if(l.le.1) go to 2745
        write(42,900)
```

```
          nlpo=nlpo+1
2745      write(42,2760) mout(k),mout(kk),knm(i,l)
2760      format("+",32x,a8,a6,10x,i4)
2790      continue
2820      write(42,150)
          nlpo=nlpo+3
          nlpo=mod(nlpo,60)
          if((nlpo+9).lt.61) go to 2835
          write(42,1185)
          nlpo=0
c
c
c********* PRINTS OUT OPERATIONS INFORMATION ON:  STATIONS, TRACKLINES, AND
c             DEPLOYMENTS *********
c
c ---- STATION DATA: STATIONS OCCUPIED AND TOTAL TIME ON STATION ----
2835      write(42,2850) dsh,dsh,dsh,dsh
2850      format(" ",2a8,"------"/" OPERATIONS INFORMATION"/" ",2a8,"------"
          )
          if(lstat) go to 2875
          if(.not.lsad) go to 2910
2875      hours=custa/60.
          write(42,2880) ksta,hours,und
2880      format("0",3x,"STATION DATA",4x,"STATIONS OCCUPIED: ",i3,",",3x,
          "TOTAL TIME ON STATION: ",f5.1," HRS,"/"+",3x,a8,"____")
2910      if(itrk.eq.0) go to 3000
c
c ---- TRACKLINE DATA: NO. OF TRACKLINES RUN, TOTAL TRACKLINE TIME AND CUMULA-
c        TIVE TRACKLINE DISTANCE ----
          hours=cugeo(itrk,1)/60.
          write(42,2940) und,medk(itrk,1),hours
2940      format("0",3x,"TRACKLINES"/"+",3x,a8,"__",t24,"TRACKLINES RUN: ",
          i3,",",3x,"TOTAL TRACKLINE TIME:  ",f5.1," HRS,")
          if(lnv) go to 2950
          nauts=5.0*hours
          sysdx(itrk,1)=nauts/.5399568
          go to 2960
2950      nauts=sysdx(itrk,1)*.5399568
2960      write(42,2970)sysdx(itrk,1),nauts
2970      format("0",16x,"CUMULATIVE TRACKLINE DISTANCE:  ",f6.1," KM / ",
          f6.1," N. MILES")
          if(.not.lnv) write(42,1940) rnot
3000      if(nd.eq.0) go to 3180
c
c ---- DEPLOYMENTS: GEOPROBES, SONOBUOYS, ON-BOTTOM-SEISMOMETERS, ETC. ----
          write(42,3030) und
3030      format("0",3x,"DEPLOYMENTS"/"+",3x,a8,"___")
          write(42,900)
          ih=0
          do 3150 i=1,nd
          k=2*ide(i)-1
          kk=k+1
          ih=ih+1
          if(ih.eq.1) write(42,3060) dout(k),dout(kk),kdep(i)
          if(ih.eq.2) write(42,3090) dout(k),dout(kk),kdep(i)
          if(ih.eq.3) write(42,3120) dout(k),dout(kk),kdep(i)
3060      format(" ",5x,a8,a6,i4)
3090      format("+",26x,"/ ",a8,a8,i4)
3120      format("+",49x,"/ ",a8,a8,i4)
          if(ih.eq.3) ih=0




3150      continue
3180      return
          end
```

```
COMPILATION LISTING OF SEGMENT dist
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1643.2 pst Tue
    Options: table symbols

c********* DISTANCE CALCULATION *********
          subroutine dist(rltn2,rlnn2,nvd2,nvh2,rmnv2,vav)
          logical lbadnv,ldis,lfst,lin,lps9,lsam,lshnt,lst,lstint
          double precision sd,se,cd,ce,cf,c,d,e,f,g,x
          dimension tm(10),dm(10)
1         common/dis/rltn1,rlnn1,rmnv1,tsst,test,tmc,dxc,dx,istint,nvd1,
          nvh1,nv,nw,lbadnv,ldis,lfst,lps9,lsam,lshnt,lst,lstint
          darcos(x)=1.570796327-datan(x/dsqrt(1.0d0-x**2))
          data c/1.745329d-2/
c
c ---- IF SUCCESSIVE RECORDS HAVE NAVIGATION, BRANCH TO 30 WHERE DISTANCE
c        BETWEEN RECORDS IS CALCULATED DIRECTLY ----
          lbadnv=.false.
c ---- ldis = f => LAST DATA RECORD DID NOT HAVE NAVIGATION ----
          if(ldis) go to 30
          dx=0.
          go to 120
c
c
c ---- SPHERICAL TRIG FUNCTIONS CALCULATING dx ON SFC OF A SPHERE USING 6368.7
c        KM AS MEAN RADIUS OF EARTH ----
30        d=rltn1*c
          e=rltn2*c
          f=c*(rlnn2-rlnn1)
          sd=dsin(d)
          se=dsin(e)
          cd=dcos(d)
          ce=dcos(e)
          cf=dcos(f)
          g=sd*se+cd*ce*cf
          if(dsqrt(1.0d0-g**2).eq.0) go to 110
c ---- DX STORES DISTANCE INCREMENTED BETWEEN SUCCESSIVE RECORDS ----
          dx=darcos(g)*6368.7
c ---- tmd STORES TIME DIFFERENCE (MINUTES) BETWEEN SUCCESSIVE RECORDS WHICH
c        HAVE NAVIGATION ----
60        tmd=1440*(nvd2-nvd1)+60*(nvh2-nvh1)+rmnv2-rmnv1
c
c
c ---- NEXT 12 STATEMENTS CALCULATE THE AVERAGE SHIP VELOCITY BASED AT A
c        GIVEN MOMENT (EXCEPT AT BEGINNING AND END OF NAVIGATION COVERAGE) ON
c        THE PREVOUS 10 RECORDS.  THIS WILL BE USED TO ESTIMATE DISTANCE FOR
c        THOSE TIME INTERVALS THAT HAVE NO NAVIGATION ----
          if(tmd.eq.0) go to 110
c ---- CALCULATES VELOCITY (KM/HR) FOR EACH dx INCREMENT ----
          vel=dx/(tmd/60.)
          if(vel.lt.31) go to 80
          if(vel.gt.60) go to 65
          if(lshnt) go to 70
c ---- lbadnv = t => NAVIGATION FOR THIS RECORD IS > 60 KM/HR AND VERY
c        QUESTIONABLE. 'DX' INCREMENT CALCULATED FROM IT WILL NOT BE USED ----
65        lbadnv=.true.
          lshnt=.true.
          go to 110
70        lshnt=.false.
          go to 110
```

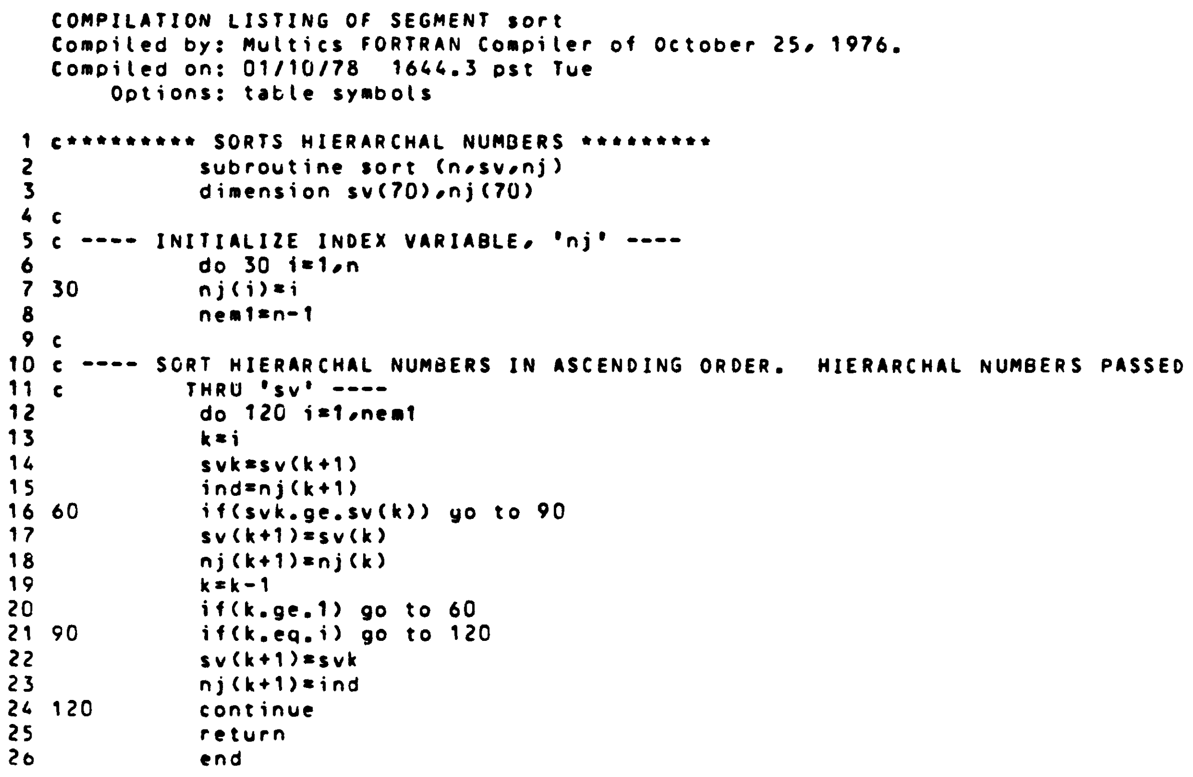
```
c
c ---- CALCULATES MOVING AVERAGE VELOCITY BASED ON PREVIOUS 10 RECORDS  ----
80        nv=nv+1
          dxc=dxc+dx
          tmc=tmc+tmd
c ---- lps9 => 10 dx INCREMENTS HAVE BEEN CALCULATED ----
          if(lps9) go to 90
          if(nv.lt.11) go to 100
          lps9=.true.
90        nw=nw+1
c ---- SUBTRACTS THE PREVIOUSLY CALCULATED TIME AND dx INCREMENT FROM
c        CUMULATIVE TIME AND dx ----
          tmc=tmc-tm(nw)
          dxc=dxc-dm(nw)
c ---- RESETS nv TO RE-USE ARRAY SPACES IN tm AND dm ----
          if(nv.gt.10) nv=1
100       tm(nv)=tmd
          dm(nv)=dx
c ---- CALCULATES AVERAGE VELOCITY ----
          tmch=tmc/60.
          vav=dxc/tmch
          if(nw.gt.9) nw=0
110       return
c
c
c        THIS SECTION DEALS WITH RECORDS WHICH HAVE NO NAVIGATION.  TIME ON
c        STATION IS SUBTRACTED OUT AND THE RESULT IS MULTIPLIED BY PREVIOUSLY
c        DETERMINED AV. VELOCITY TO GIVE AN ESTIMATE OF THE DX BETWEEN THESE
c        RECORDS ----
120       tmp=0
c ---- CONVERTS TIME OF 1ST RECORD W/NAV TO DECIMAL MINUTES (tnv1) ----
          tnv1=1440*nvd1+60*nvh1+rmnv1
c ---- lst = t => RECORD IS 'START' OF A STATION RECORD ----
          if(lst) go to 170
c ---- CONVERTS TIME OF 2ND RECORD W/NAV TO DECIMAL MINUTES (tnv2) ----
          tnv2=1440*nvd2+60*nvh2+rmnv2
c ---- lsam = t => CURRENT RECORD IS A SAMPLE RECORD ----
          if(.not.lsam) go to 130
c ---- lfst = t => CURRENT RECORD IS 1ST SAMPLE ENCOUNTERED ON A PARTICULAR
c        STATION ----
          if(.not.lfst) go to 180
c ---- lstint= TRUE => RECORD IS W/IN STATION INTERVAL AS DETERMINED BY
c        RECORDS, NOT STATION RECORDS ----
          lstint=.true.
c ---- tmp IS TIME DIFFERENCE BETWEEN 'FIRST' RECORD W/NAV AND START OF LAST
c        STATION ENCOUNTERED ----
          tmp=tsst-tnv1
c ---- tsst < tnv1 => SHIP DEAD IN WATER SINCE FIRST RECORD W/NAV.  THEREFORE,
c        dx IN STATEMENT 210 WILL BE ZERO ----
          if(tsst.lt.tnv1) tmp=0
          go to 180
130       if(.not.lstint) go to 160
          istint=istint+1
          if(istint.gt.8) go to 154
c ---- QUERIES WHETHER NON-SAMPLE RECORD EVENT TOOK PLACE ON A STATION ----
          write(6,138) nvd2,nvh2,rmnv2
138       format("0 NON-SAMPLE RECORD OF DAY TIME: ",i2,"/",i2,f4.1," FOLL",%
          "OWS A SAMPLE RECORD FALLING IN A NAVIGATION GAP."/"   IS THIS R",%
          "ECORD WITHIN THE TIME SPAN OF THE STATION AT WHICH SAMPLE COLLE",%
          "CTED?? (t OR f)")
```

```
        read(5,146) lin
146     format(l1)
        if(.not.lin) go to 154
        go to 180
154     lstint=.false.
        istint=0
c ---- test = END OF STATION TIME. test < tnv2 => VELOCITY FOR tnv2-tnv1 is > ZERO ----
160     if(test.ge.tnv2) go to 180
        tmp=tnv2-tnv1
        go to 180
c ---- tsst < OR = tnv1 => VELOCITY FOR TIME INTERVAL BETWEEN FIRST AND
c       LAST RECORDS WITH NAV WAS ZERO ----
170     if(tsst.le.tnv1) go to 180
        tmp=tmp+(tsst-tnv1)
180     tmd=tmp/60.
c ---- ESTIMATE OF DISTANCE BETWEEN SUCCESSIVE RECORDS BASED ON TIME INTERVAL
c       BETWEEN THEM AND LAST DETERMINATION OF THE AVERAGE VELOCITY ---
210     dx=vav*tmd
240     return
        end
```

```
COMPILATION LISTING OF SEGMENT datcon
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1643.6 pst Tue
    Options: table symbols

c********* DATE CONVERSION *********
          subroutine datcon (mon,dyl,hrl)
          integer dyl,dyjdl,edmo,edmol,hrl,s
          common/stprdt/idc(30,6),jdc(30),ihc(30),kc,s(20)
          dimension edmo(11),edmol(11),mon(30),dyl(30),hrl(30)
c
c ---- DATA STATEMENTS 'edmo' AND 'edmol' CONTAIN JULIAN DAYS FOR THE LAST
c        DAY OF EACH MONTH FOR NORMAL AND LEAP YEARS ----
          data edmo/31,59,90,120,151,181,212,243,273,304,334/, edmol/31,60,
          91,121,152,182,213,244,274,305,335/
c
c
c ---- LOOP CONVERTS JULIAN DAY AND GMT TO MONTH OF YEAR AND A LOCAL STANDARD
c        TIME BASED ON THE NUMBER OF 15 DEG. INCREMENTS FROM THE GREENWICH
c        MERIDIAN ----
          do 300 i=1,kc
          mn=1
          kedmol=0
          if(i.eq.1) go to 30
c ---- MODULO FUNCTION WHICH SHUNTS 'START' OF CRUISE OR PORT STOPS TO
c        SECTION WHICH PROMPTS FOR PORT MERIDIANS ----
          if(mod(i,2).ne.0) go to 180
          go to 75
c
c ---- READS PORT MERIDIAN AND DETERMINES HOW MANY 15 DEG. INTERVALS IT IS
c        AWAY FROM GREENWICH MERIDIAN ----
30        write(6,60)
60        format(" ENTER THE LONGITUDINAL MERIDIAN OF THE PORTS WHERE SHIP",%
          " BEGINS CRUISE, WHERE IT STOPS MID-CRUISE,"/"   AND WHERE IT EN",%
          "DS CRUISE.  USE DECIMAL DEGREES W/OUT (-) OR (+) PREFIX. E.G., ",%
           "163.1234.")
75        write(6,90) (idc(i,j),j=1,6)
90        format(" ENTER MERIDIAN FOR PORT:  ",6a4,"  ,AFTER>>")
          write(6,120)
120       format(" >>")
c ---- plon CONTAINS PORT MERIDIAN ----
          read(5,150) plon
150       format(f8.4)
          rtm=plon/15.0
          itim=ifix(rtm)
c ---- ROUND OFF TO NEXT HIGHER DEGREE IF DECIMAL PORTION > .5 DEGREES ----
          rztm=itim+.5
          if(rtm.ge.rztm) itim=itim+1
c ---- ADJUST GMT HOUR READING IN 'ihc' TO LOCAL STANDARD HOUR ----
180       hrl(i)=ihc(i)-itim
          if(hrl(i).lt.0) go to 210
          dyjdl=jdc(i)
          go to 240
c ---- ADJUST JULIAN DAY BACK ONE DAY IF LOCAL TIME IS EARLIER THAN MIDNIGHT ----
210       dyjdl=jdc(i)-1
c ---- ADJUST HOUR FROM A NEGATIVE TIME TO A POSITIVE TIME ----
          hrl(i)=hrl(i)+24
c ---- SPECIAL CASE IF JULIAN DAY IS IN MONTH OF JANUARY ----
240       if(dyjdl.lt.32) go to 280
c ---- rout IS ESTIMATE OF CURRENT MONTH ----
```

```
            rout=dyjdl/30.42
            mn=ifix(rout)
            rmon=mn+.5
c ---- INCREASES MONTH KEY TO NEXT MONTH ----
            if(rout.ge.rmon) mn=mn+1
c ---- MODULO FUNCTION WHICH SENDS LEAP YEAR RECORDS TO 270 ----
            if(moa(s(14),4).eq.0) go to 270
c ---- INCREASES MONTH KEY TO NEXT MONTH IF LOCAL JULIAN DAY > LAST DAY OF CURRENT
c         MONTH ----
            if(dyjdl.gt.edmo(mn)) mn=mn+1
c ---- SUBTRACTS LOCAL JULIAN DAY FOR LAST DAY OF PREVIOUS MONTH FROM CURRENT
c         LOCAL JULIAN DAY TO GIVE DAY INTO CURRENT MONTH ----
            dyl(i)=dyjdl-edmo(mn-1)
            go to 300
270         if(dyjol.gt.edmol(mn)) mn=mn+1
            kedmol=edmol(mn-1)
280         dyl(i)=dyjdl-keamol
300         mon(i)=mn
            return
            end
```



```
COMPILATION LISTING OF SEGMENT sort
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1644.3 pst Tue
    Options: table symbols

c********* SORTS HIERARCHAL NUMBERS *********
          subroutine sort (n,sv,nj)
          dimension sv(70),nj(70)
c
c ---- INITIALIZE INDEX VARIABLE, 'nj' ----
          do 30 i=1,n
30        nj(i)=i
          nem1=n-1
c
c ---- SORT HIERARCHAL NUMBERS IN ASCENDING ORDER.  HIERARCHAL NUMBERS PASSED
c        THRU 'sv' ----
          do 120 i=1,nem1
          k=i
          svk=sv(k+1)
          ind=nj(k+1)
60        if(svk.ge.sv(k)) go to 90
          sv(k+1)=sv(k)
          nj(k+1)=nj(k)
          k=k-1
          if(k.ge.1) go to 60
90        if(k.eq.i) go to 120
          sv(k+1)=svk
          nj(k+1)=ind
120       continue
          return
          end
```

```
COMPILATION LISTING OF SEGMENT blockdata
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/10/78  1644.7 pst Tue
     Options: table symbols

c********* BLOCK DATA *********
            block data
            character*1 shp
            character*2 b2,shpp
            character*3 month
            character*4 area,type
            character*8 depl,dot1(119),dot2(119),dot3(78),dout,dsh,loc,mout,ship,und
            integer are
c
c ---- LABELED COMMON STATEMENTS CONTAINING VARIABLES COMMON TO 'PRINT' ----
            common/blok/dout(316),dsh,loc(5),mout(30),ship(26),und,
            are(10),area(50),type(12),month(12),shp(10),shpp(3),b2
            equivalence (dout(1),dot1(1)),(dout(120),dot2(1)),(dout(239),dot3(
            1))
c
c ---- DATA STATEMENTS CONTAINING AREA CODES, AREA DESCRIPTIONS, BLANKS, DEPLOYED
c          SYSTEMS, MONTHS, UNDERSCORES, SHIP CODES, SHIP NAMES, AND
c          GENERAL DATA CATEGORIES ----
            data are/"BS","EG","NC","WG","SC","WO","AR","MX","NP","HW"/,
            area/"BERI","NG S","EA  ","    ","    ","E. G","ULF ","OF A",
            "LASK","A   ","N. C","ALIF","ORNI","A CO","AST ","W. G","ULF ",
            "OF A","LASK","A   ","S. C","ALIF","ORNI","A CO","AST ","WASH",
            "./OR","EGON"," COA","ST  ","ARCT","IC O","CEAN","    ","    ",
            "COAS","T OF"," MEX","ICO ","    ","N. P","ACIF","IC O","CEAN",
            "    ","HAWA","IIAN"," ISL","ANDS","    "/,b2/"  "/,
            dsh/"--------"/,loc/5*"        "/,month/"JAN","FEB","MAR","APR",
            "MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"/,und/"________"/,
            shp/"S","G","M","L","K","V","O","T","A","D"/,shpp/"SC","GL","ME"/,
            ship/"SEA SOUN","DER     ","GROWLER ","        ","JOHN J. ",
            "MILLER  ","SAMUEL P",". LEE   ","KARLUK  ","        ","VELERO  ",
            "        ","OCEANOGR","APHER   ","TOMMY TH","OMPSON  ","ACONA   ",
            "        ","DISCOVER","ER      ","ELLEN B."," SCRIPPS","GLACIER ",
            "        ","MELVILLE","        "/,type/"GEOP","HYSI","CAL ",
            "GEOL","OGIC","AL  ","HYDR","OGRA","PHIC","    ","    ","    "/
c
c ---- DATA STATEMENT CONTAINING MEDIA OR GENERAL DATA GROUP DESCRIPTORS ----
            data mout/"        ","        ","PERSONNE","L LIST","EQUIPMEN",
            "T LIST","SAMPLE A","TTEMPT","ANL PAPE","R ROLL","DIGIT MA",
            "G TAPE","OPRTING ","PERIOD","NUM.OBSR","VATION","ANLOG MA",
            "G TAPE","    PHOT","OGRAPH","PRINTR L","ISTING","    DEPL",
            "OYMENT","   MAP O","R PLOT","COMP CAR","C DECK","DIG PUNC",
            "H TAPE"/
c
c ---- DATA STATEMENT DOUT (EQUIVALENT TO dot1, dot2, dot3) WHICH CONTAINS A
c          DATA TYPE DESCRIPTION KEYED TO DATA CODES IN 'COD' FROM EDITCDI.  KEY
c          IS jc OR ic ----
            data dot1/"CRUISE  ","        ","IN PORT ","        ","SHIP CAP",
            "TAIN    ","CHIEF EN","GINEER  ","CHIEF MA","TE      ","CHIEF SC",
            "IENTST  ","DAFE CUR","ATOR    ","GEOLOGIS","T       ","GEOPHYSI",
            "CIST    ","GEOCHEMI","ST      ","BIOLOGIS","T       ","OCEANOGR",
            "APHER   ","ELECTRON","ICS T   ","MECHANIC","AL T    ","WATCH ST",
            "ANDER   ","NAVIGATO","R       ","UNSP INV","ESTIGATR","        ",
            "        ","        ","        ","NAV SATE","LLITE   ","DOPPLER ",
            "SONAR   ","RADAR   ","        ","MINIRANG","ER      ","DEL NORT",
            " RNG/RNG","INTEGRAT","ED NAV  ","LORAN A ","        ","LORAN C ",
      "        ","LORAN C ","RHO-RHO ","HI-FIX  ","        ","RAY DIST",
      "        ","COURSE R","ECORDER ","LORAC NA","V       ","        ",
      "        ","        ","        ","DR NAV P","LOT/LAB ","DR NAV P",
      "LOT/ERIG","DR NAV P","LOT/COMP","     TRA","CKLINE  ","       S",
      "TATION  ","        ","        ","        ","        ","        ",
      "        ","AIRGUN  ","        ","SPARKER ","        ","24 CHANL",
      " SEISMIC","36 CHANL"," SEISMIC","48 CHANL"," SEISMIC","72 CHANL",
      " SEISMIC","        ","        ","        ","        ","MULT CHA",
      "N MONITR","SNGL CHA","N AIRGUN","SNGL CHA","NL ARCER","S CHA AI",
      "RGN/ARCR","        ","        ","BURG AIR","GUN     ","BURG SPA",
      "RKER    ","BURG AIR","GN/SPRKR","BOTTOM S","EISMOMTR","SONOBUOY"/
      data dot2/"        ","EXPLOSIV","ES      ","UNIBOOM ","        ",
      "MINISPAR","KER     ","DEL NORT","E       ","SIDE SCA","N SONAR ",
      "UNIBOOM/","MINSPRKR","        ","        ","DIGITRAC","K       ",
      "3.5KH BA","THYMETRY","7 KHZ BA","THYMETRY","12KH BAT","HYMETRY ",
      "48KH BAT","HYMETRY ","200KH BA","THYMETRY","200 + 7K","H BATHYM",
      "BATHYMET","RY + NAV","SHIPBOAR","D MAGGY ","GRADIOME","TER     ",
      "MAGGY + ","BATHYMET","MAGNETIC","S + NAV ","MAG/BATH","YM/NAVIG",
      "SHIPBOAR","D GRAVTY","GRAVITY ","+ BATHYM","GRAVITY ","+ MAGNET",
      "GRAV/MAG","/BATHYM ","GRA/MAG/","BATH/NAV","GRAV/BAT","HY NAVIG",
      "STA GRAV"," LAND TY","LAND STA","TION MAG","HEAT FLO","W GAUGE ",
      "DART COR","E       ","GRAVITY ","CORE    ","BOX(SHIP",") CORE  ",
      "PISTON C","ORE     ","VIBRATIN","G CORE  ","DRILL CO","RE      ",
      "ALPINE C","ORE     ","BOOMERAN","G CORE  ","HAND BOX"," CORE   ",
      "PHLEGER ","CORE    ","HYDROPLA","ST. CORE","        ","        ",
      "VAN VEEN"," GRAB   ","SHIPECK ","GRAB    ","FREEFALL"," GRAB   ",
      "PETERSON"," GRAB   ","PONAR GR","AB      ","CAMPBELL"," GRAB   ",
      "ORANGE P","EEL GRAB","HAND SAM","PLE GRAB","SOUTAR G","RAB     ",
      "DIETZ/LF","OND GRAB","        ","        ","CHAIN DR","EDGE    ",
      "PIPE DRE","DGE     ","UNDERWAY"," SAMPLER","SUSPEND ","SEDIMENT",
      "TELEVISI","ON      ","SEAFLOOR"," CAMERA ","        ","        "/
      data dot3/"SCUBA DI","VE      ","LAND SAM","PLE     ","PENETROM",
      "ETER    ","SIDE SCA","N MARKER","NISKEN B","OTTLE   ","NANSEN B",
      "OTTLE   ","VAN DORN"," BOTTLE ","WATER BU","CKET    ","        ",
      "        ","        ","        ","        ","        ","TRANSMIS",
      "SOMETER ","NEPHELOM","ETER    ","SECCHI D","ISK     ","        ",
      "        ","        ","        ","AND CURR","ENT METR","GEO CURR",
      "ENT METR","BEN CURR","ENT METR","MAR CURR","ENT METR","HYD CURR",
      "ENT METR","        ","        ","        ","        ","ANDERA T",
      "IDE METR","        ","        ","        ","        ","WAVE PRE",
      "S SENSOR","EXP BATH","Y THERMO","TEMP/SAL","INOMETER","PLANKTON",
      " NET    ","TEMPRATU","RE GAUGE","RIPPLE P","ROFILER ","        ",
      "        ","BECKMAN ","STC     ","CTD METE","R       ","GEOPROBE",
      "        ","        ","        ","        ","        ","        ",
      "        "/
      end
```

PROGRAM PLOTCDI

```
PROGRAM PLOTCDI - PLOT CRUISE DATA INDEX
  APPLICATION   - SPECIFIC TO CRUISE DATA INDEX
  FUNCTION      - SELECTS NAVIGATION RECORDS FROM DATA RECORDING
                    INTERVALS - RECORDS USED TO PLOT DATA LOCATIONS
  MODE          - INTERACTIVE


==============================================================================
                          I D E N T I F I C A T I O N
==============================================================================

  PROGRAM.......PLOTCDI

  DESCRIPTION...THIS PROGRAM SELECTS NAVIGATION RECORDS FROM LARGER
                NAVIGATION FILES OVER THE RECORDING INTERVALS OF
                SPECIFIC UNDERWAY DATA SYSTEMS.  RECORDING INTERVALS
                OF A DATA SYSTEM ARE BRACKETED BY CRUISE DATA INDEX
                RECORDS CONTAINING THE START, END, ON, OR OFF TIMES
                FOR THE SYSTEM.  THE SELECTED NAVIGATION RECORDS CAN
                THEN BE USED WITH THE PLOTTING PROGRAM NAVPLT TO PLOT
                THE LOCATION TRACK OF THE UNDERWAY DATA.  SPECIFIC
                CAPABILITIES OF PLOTCDI INCLUDE:
                (1) TESTS WHETHER A GIVEN NAVIGATION RECORD FALLS
                    BETWEEN SUCCESSIVE CDI RECORDS BRACKETING A
                    RECORDING INTERVAL FOR A PARTICULAR DATA  SYSTEM.
                (2) WRITES ALL NAVIGATION RECORDS FALLING WITHIN A
                    DATA RECORDING INTERVAL TO A SEPARATE FILE IN THE
                    USER'S WORKING DIRECTORY IN BINARY FORMAT.  THIS
                    IS THE PLOT FILE.  ALSO DUPLICATES THE FILE IN THE
                    WORKER'S DIRECTORY UNDER THE NAME:  FILE13, AS A
                    FORMATTED ASCII FILE WHICH CAN BE VISUALLY
                    CHECKED.
                (4) WRITES DUMMY NAVIGATION RECORDS TO THE OUTPUT FILE
                    WHICH SIGNAL TO NAVPLT WHEN TO LIFT THE PLOTTING
                    PEN FROM THE PLOT PAPER AS WELL AS WHEN TO CHANGE
                    PLOT SYMBOLS FOR THE DIFFERENT CRUISES ENCOUNTERED
                    IN A PROGRAM RUN.

  VERSION....... 6 JAN 1978

  CONTACT.......TOM CHASE, BRADLEY LARSEN
                MARINE GEOLOGY, U.S. GEOLOGICAL SURVEY, MENLO PARK

  PROGRAMMING...B. LARSEN  ORIGINAL                             NOV 1977
                    BASE PROGRAM
                B. LARSEN  MODIFICATION                         JAN 1978
                    PATCH IN G. McHENDRIE'S MULTI FILE SUBROUTINES
```

PROGRAM PLOTCDI

```
COMPUTERS.....HONEYWELL SERIES 60 UNDER MULTICS
PROCESSORS....HONEYWELL MULTICS FORTRAN (RELEASE 4)

INPUT.........(1) ANSWERS TO INTERACTIVE PROMPTS (TERMINAL)
              (2) FILES CONTAINING CDI FORMATTED ASCII RECORDS OF A
                  SINGLE DATA TYPE WHICH HAVE BEEN PRE-SELECTED FROM
                  ONE OR MORE CRUISE DATA INDICES.  THE FILES ARE
                  READ BY PLOTCDI FROM THE USER'S WORKING DIRECTORY
                  OR TAPE (UNIT 08)
              (3) NAVIGATION DATA FILES EXISTING AS:  FORMATTED
                  ASCII RECORDS;  UNFORMATTED BINARY RECORDS USING A
                  MAGNAVOX VARIABLE LIST;  OR UNFORMATTED BINARY
                  RECORDS USING A MARCONI VARIABLE LIST.  FILES IN
                  ANY OF THESE FORMATS MAY BE ACCESSIBLE FROM EITHER
                  THE USER'S WORKING DIRECTORY OR FROM STANDARD
                  LABEL ASCII TAPES.  (UNIT 10)

OUTPUT........(1) FILE (NAMED BY USER) WRITTEN TO USER'S WORKING
                  DIRECTORY AND CONTAINING RECORDS IN BINARY FORMAT
                  OVER THE RECORDING INTERVALS OF A PARTICULAR DATA
                  SYSTEM.  USED FOR PLOTTING.
              (2) DUPLICATE OF (1), ONLY IN A READABLE ASCII
                  CHARACTER FORMAT (NAMED FILE13).  USED FOR VISUAL
                  CHECKING.

SUBROUTINES...MULTI_FILE,MF_INIT   PERFORM ATTACH, OPEN, CLOSE AND
                         DETACH OPERATIONS ON SUCCESSIVE DISK OR
                         TAPE FILES.

EXECUTION.....PROGRAM CAN BE RUN BY INVOKING: ec plot
```

OPERATION

EXAMPLE TYPE

The following example is given to show how to run PLOTCDI using two Cruise Data Index segments and two navigation files as input. The CDI segments consist of the start/end/on/off records for 3.5 khz Bathymetry which were pre-selected from the Cruise Data Indices of two separate cruises using the subroutine SELSCH of the program EDITCDI. The segments are named sea276.35cdi and sea376.35cdi and reside in the disk storage area of the Honeywell Multics system known as the user's 'working directory' which makes them directly accessible to the user's programs. The navigation files used in the example consist of 'cleaned' dead reckoning lat/long fixes which cover the entire time periods of each cruise. In this particular example, they reside on a single magnetic tape in two separate files and in a binary format peculiar to the magnavox sat-nav system and the file names are F2 and F3 and they reside on the 2nd and 3rd file of the tape.

PRECAUTIONS

----

(1) CDI input is limited to 10 segments in a single program run. They are read by unit 08 which utilizes the standard CDI 117 ascii character record format. CDI input files may reside on tape or in the user's working directory but in general it will be more convenient to access them from the user's directory because that is where they are created by the subroutine SELSCH of the program EDITCDI when they are selected from the larger CDI files. Wherever they reside, the files can be accessed from only one device during a given program run.

(2) Navigation input is also limited to 10 input files in a program run. However, the files can be accessed from more than one storage device and can be in 3 separate formats as mentioned in the output section in the program 'IDENTIFICATION' section above. Any one storage device can have more than one navigation file but the user can only access files of the same format from the device in a single pass through the program. Files in a different format on the same device can be accessed only by initiating further passes through the program. This re-initiates mf_init and multi_file which re-attach and re-open the device as if it had been encountered for the first time.

(3) CDI files which overlap in time should not be run back-to-back. Those that do overlap should be separated by files for another cruise or should be accessed in separate program runs.

(4) The assumption is made by PLOTCDI that each separate input CDI file/segment represents a different cruise and has a corresponding navigation file.

(5) Segments containing the I/O control statements required by the multi file subroutines in PLOTCDI should be created in the user's directory prior to a program run. No more than two input control records should be placed in each segment and only a single control record need be created for all input CDI files if the CDI files exist entirely in the user's directory or on a single device. In general, the first control segment will have one record for all of the CDI files used in the program run and one record for the navigation files on the first device accessed. Further control segments will contain only one input control record for navigation files that exist on a different device or in a different format on the same device. An example of a

control segment is shown:

```
CDI input:  in -sw file08 -nb . -nm sea276.35cdi,sea376.35cdi <vfile_ s
NAV input:  in -sw file10 -nb 2,3 -nm . <tape_ansi_ S6NCL3 -ret all>
            end
```

Only the file names (sea276.35cdi, etc) of the CDI input control record need be changed for different program runs unless subsequent CDI files are on a device other than the working directory. They must be entered after -nm in the order they are to be accessed and with no imbedded blanks between them. In general, the file numbers (2,3), and tape name (S6NCL3) are the only items that need changing when making up a control record for a new navigation storage device. An alternative method of naming CDI or navigation files is to use the convention of name1, name2. When this is done, only the entry: name(1) need be entered in the file-name slot. The first file read will be taken as name1; subsequently, the number in parentheses will be incremented by one and the program will search for files name2, name3.....namei+1
A control segment is linked to the program by typing: get control_seg_name, after the user is asked to 'Enter I/O control statements:' at the beginning of a program run. For further information on multi file I/O see Graig McHendrie's documentation on the mf_init and multi_file subroutines.

(6) In the example, interactive prompts or messages to the terminal are prefaced by a dash (-); system responses are prefaced by a bracket ([).

PROGRAM RUN

```
ec plot
 -Enter switch number, eg, 09
 -10
 -Enter name for dummy segment (max. 6 chars):
 -dum
 -Input or output action (i/o)?
 -i
 -ENTER OUTPUT SEGMENT NAME - UP TO 32 CHARACTERS.
sea2376.35nav
 -??HOW MANY CDI FILES ARE BEING PROCESSED?? (i2)
02
 -??HOW MANY NAV FILES FROM THE FIRST DEVICE ARE BEING
   PROCESSED?? (i2)
02
 -PROMPTS FOR INPUT CONTROL STATEMENTS FOR DEVICES CONTAINING
   CDI'S AND CORRESPONDING NAVIGATION FILES.
 -ENTER ONLY ONE STATEMENT FOR EACH DATA TYPE AT THIS TIME.
   SWITCH NAMES:  NAVIGATION, file10;  CDI CARDS OR TAPE,
```

PROGRAM PLOTCDI

```
                   file08
 -Enter I/O control statements:
get control
 -??CURRENT NAVIGATION IS ON WHAT STORAGE MEDIUM??
   ENTER: cards, magna, or marco - FOR CARDS, MAGNAVOX, OR
   MARCONI TAPE.
magna
 -file08 attach vfile_ sea276.35cdi
 -file08    opened si
 -file10 attach tape_ansi_ S6NCL3 -ret all -nb 2 -nm F2
 [Mounting volume S6NCL3 with no write ring.
 [S6NCL3 mounted on tape_01
 -file10    opened
 -END OF NAV FILE ON UNIT 10.  NUMBER OF RECORDS READ = 3149
   FROM FILE NUMBER   1
 -file10    closed   detached
 -file10 attach tape_ansi_ S6NCL3 -ret all -nb 3 -nm F3
 -file10    opened
 -END OF CDI FILE ON UNIT 08.  NUMBER OF RECORDS READ =  96
   FROM FILE NUMBER   1
 -file08    closed   detached
 -file08 attach vfile_ sea376.35cdi
 -file08    opened  si
 -END OF CDI FILE ON UNIT 08.  NUMBER OF RECORDS READ = 67
   FROM FILE NUMBER   2
 -END OF JOB
 -TOTAL CDI RECORDS READ = 163
 -TOTAL NAV RECORDS READ = 6612
 -STOP
 [fortran_io_: Close files?   yes
```

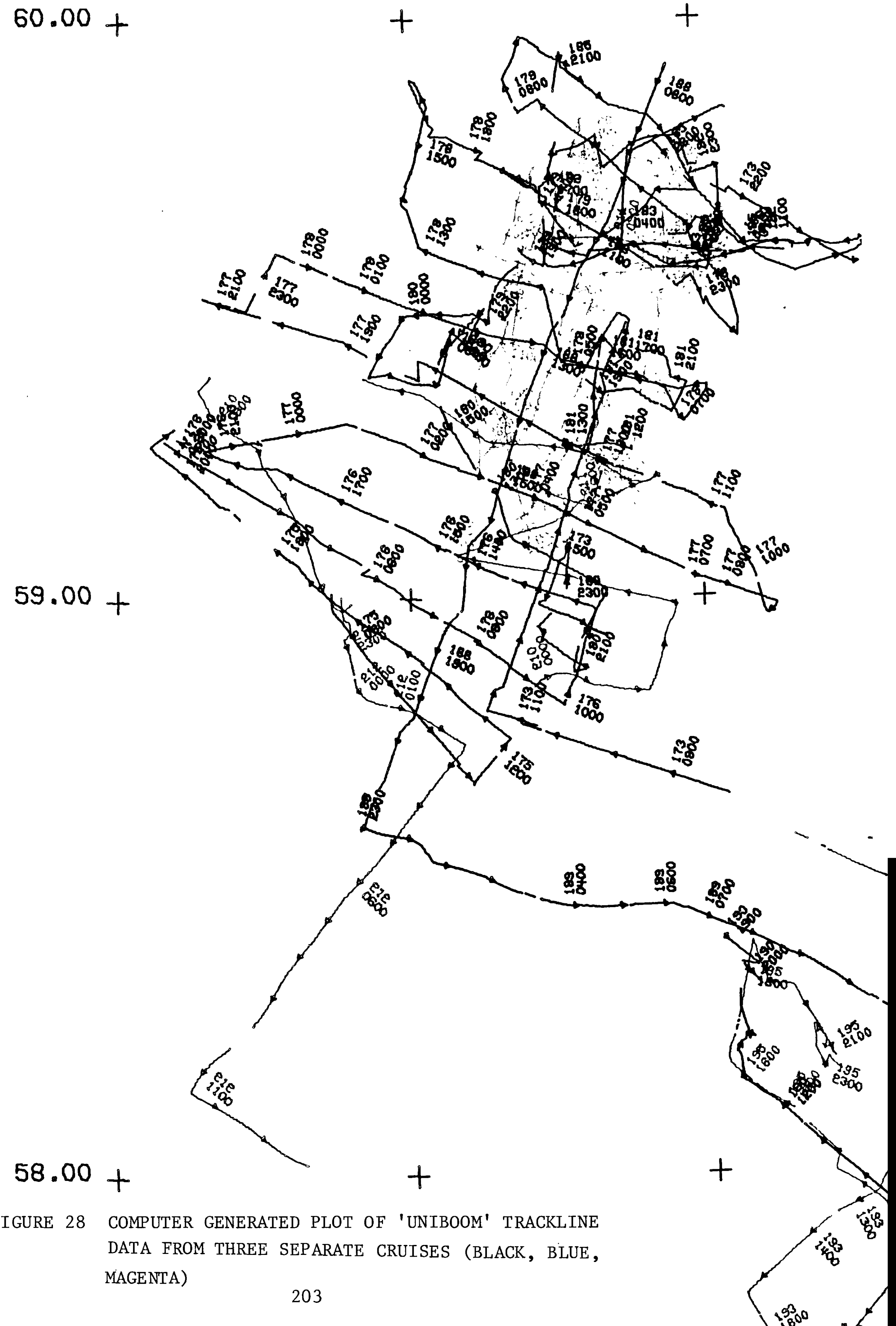


FIGURE 28 COMPUTER GENERATED PLOT OF 'UNIBOOM' TRACKLINE DATA FROM THREE SEPARATE CRUISES (BLACK, BLUE, MAGENTA)

COMPILATION LISTING OF SEGMENT plotcdi
Compiled by: Multics FORTRAN Compiler of October 25, 1976.
Compiled on: 01/16/78 1508.7 pst Mon
    Options: table symbols

```
c********* PLOTCDI *********
           logical ioeror,lbcdi,lfnav,lmcdi,lnav
           character*4 idd1(6),idd2(6)
           character*5 nt(3),ntyp
           character*7 ordnl(10)
           character*32 segout
           double precision dam,dcse,dlat,dlon,dsdlat,dsdlon,dspd,dumya8
           integer dy1,dy2,hr1,hr2,ic1,ic2,im1,im2,iu1,iu2,s1(20),s2(20),sp1,
           sp2,nnin,ncin,ncdi,nnav
           external mf_init,multi_file
           data nt/"cards","magna","marco"/, ordnl/"FIRST  ","SECOND ",
           "THIRD  ","FOURTH ","FIFTH  ","SIXTH  ","SEVENTH","EIGHTH ",
           "NINTH  ","TENTH  "/,nvdh,nvhh,ncdi,nnav,nnrin,ncrin,nntot,nctot,
           rltd,rlnd,rmnh,kn,rmnd,ioeror,lbcdi,lfnav,lmcdi,lnav/8*0,
           3*0.,1,-2.0,5*.false./
c
c
c ---- PROMPTS FOR OUTPUT DATASET NAME AND ATTACHES SWITCH NAME TO OUTPUT
c         DATASET ----
           write(6,30)
30         format(" ENTER OUTPUT NAVIGATION SEGMENT NAME - UP TO 32 CHARACTE"%
           "RS.")
           read(5,40) segout
40         format(a32)
           call io ("attach","file12","vfile_",segout,"-append")
           call io ("open","file12","sqo")
c
c ---- PROMPTS FOR INFORMATION ABOUT INPUT DATA SETS ----
           write(6,50)
50         format(" ?? HOW MANY CDI FILES ARE BEING PROCESSED?? (i2)")
           read(5,100) ncf
100        format(i2)
           write(6,150) ordnl(kn)
150        format(" ??HOW MANY NAV FILES FROM THE ",a7," DEVICE ARE BEING PR"%
           "OCESSED?? (i2)")
           read(5,100) nnf
c ---- ATTACHES SWITCH NAMES TO CDI AND NAVIGATION INPUT DATA FILES ----
           write(6,200)
200        format(" PROMPTS FOR INPUT CONTROL STATEMENTS FOR DEVICES CONTAIN"%
           "ING CDI'S AND CORRESPONDING NAVIGATION FILES."/" ENTER ONLY ONE "%
           "STATMENT FOR EACH DATA TYPE AT THIS TIME."/"   SWITCH NAMES:  NA"%
           "VIGATION, file10; CDI CARDS OR TAPE, file08")
c ---- INITIALIZE INPUT NAV AND CDI FILES ----
           call mf_init(ioeror)
           if(ioeror) go to 6000
250        write(6,350)
350        format(" ??CURRENT NAVIGATION INPUT IS ON WHAT STORAGE MEDIUM??"/
           "    ENTER: cards, magna, or marco - FOR CARDS, MAGNAVOX, OR MARCO"%
           "NI TAPE.")
400        read(5,402) ntyp
402        format(a5)
c
c ---- MATCHES NAVIGATION DATA TYPE CODE ENTERED IN ntyp WITH CODES IN DATA
c         STATEMENT ARRAY nt ----
           do 450 i=1,3
           if(ntyp.eq.nt(i)) go to 550
 450       continue
           write(6,500)
 500       format(" **ERROR**  RE-ENTER NAV. MEDIUM TYPE.")
           go to 400
 550       nb=i
c
c ---- CALLS MULTIFILE SUBROUTINE FOR INPUT SWITCH AND BRANCHES TO PROPER
c         READ STATEMENT ----
 570       call multi_file("file08",ncdi,ioeror)
           if(ioeror) go to 4000
           call multi_file("file10",nnav,ioeror)
           if(ioeror) go to 4000
           if(nb-2) 600,650,700
c
c ---- READ NAVIGATION RECORD ----
 600       read(10,752,end=1700) nvd,nvh,rmn,rltn,rlnn
 752       format(i3,1x,i2,f4.1,43x,f9.5,f10.5)
           go to 800
 650       read(10,end=1700) nvd,nvh,rmn,rltn,rlnn
           go to 800
 700       read(10,end=1700) nvd,nvh,dam,iline,ishot,itype,dlat,dsdlat,dlon,
           dsdlon,dcse,dspd,dumya8
           rltn=sngl(dlat)
           rlnn=sngl(dlon)
           rmn=sngl(dam)
 800       nnrin=nnrin+1
           if(lbcdi) go to 900
           if(lmcdi) go to 1000
c
c ---- READS IN TWO ON/OFF SEQUENCED CDI RECORDS ----
 820       read(8,850,end=1500) dy1,hr1,rmn1,lin1,sp1,isn1,(idd1(i),i=1,6),
           iu1,(s1(i),i=1,20),ic1,im1,ilt1,iln1
 850       format(i3,1x,i2,f4.1,1x,2a4,t30,a4,t41,6a4,t71,a1,t21,a3,1x,a4,a1,
           t37,4a1,t66,a1,4i1,t73,a4,i2,a2,1x,4a1,1x,i6,i4,i3,1x,i8,i9)
           ncrin=ncrin+1
           read(8,850,end=1500) dy2,hr2,rmn2,lin2,sp2,isn2,(idd2(i),i=1,6),
           iu2,(s2(i),i=1,20),ic2,im2,ilt2,iln2
           ncrin=ncrin+1
c ---- CONVERTS INPUT NAVIGATION INTEGER TO DECIMAL DEGREES ----
           rlt1=float(ilt1)/100000
           rln1=float(iln1)/100000
           rlt2=float(ilt2)/100000
           rln2=float(iln2)/100000
c
c ---- COMPARES NAV RECORD WITH 1ST CDI RECORD, STORES DIFFERENCE IN tdif1 ----
 900      lbcdi=.false.
           tdif1=1440*(dy1-nvd)+60*(hr1-nvh)+rmn1-rmn
           if(tdif1) 950,1100,970
c ---- IF NAV VALUE = ZERO, TIME/POSITION NOT WRITTEN FOR START/ON CDI
c         RECORDS ----
 950       if(rlt1.eq.0) go to 1000
c ---- lnav = true => A NAVIGATION RECORD HAS BEEN ENCOUNTERED WITH ITS TIME
c         BEFORE OR BETWEEN THE TIMES OF ANY TWO START/END CDI RECORDS ----
           if(.not.lnav) go to 1000
c
c ---- WRITES TIME AND POSITION OF 1ST CDI RECORD TO WORKING DIRECTORY IN
c         BINARY FORMAT ----
           write(12) dy1,hr1,rmn1,rlt1,rln1
         write(13,8000) dy1,hr1,rmn1,rlt1,rln1
```

```
8000   format(" ",i3,1x,i2,f4.1,43x,f9.5,f10.5)
        go to 1000
c ---- lbcdi = true => NAV RECORD < TIME FOR FIRST CDI RECORD ----
970     lbcdi=.true.
        lnav=.true.
        go to (600,650,700), nb
1000    lmcdi=.false.
c ---- tdif2 STORES TIME DIFFERENCE BETWEEN NAV RECORD AND 2ND CDI RECORD ----
        tdif2=1440*(dy2-nvd)+60*(hr2-nvh)+rmn2-rmn
        if(tdif2) 1050,1020,1100
1020    if(.not.lnav) go to (600,650,700), nb
c ---- WRITES TIME/POSITION OF 2ND CDI RECORD FOR WHEN A NAV RECORD TIME =
c       2ND CDI RECORD TIME; BINARY ----
        write(12) nvd,nvh,rmn,rltn,rlnn
       write(13,8000) nvd,nvh,rmn,rltn,rlnn
c ---- WRITES ZERO LAT/LON TO SIGNAL PLOTTER PEN TO LIFT OFF OF PAPER ----
        write(12) nvd,nvh,rmn,rltd,rlnd
       write(13,8000) nvd,nvh,rmn,rltd,rlnd
        go to (600,650,700), nb
c ---- IF NAV VALUE = ZERO, TIME/POSITION NOT WRITTEN FOR END/OFF CDI
c       RECORDS ----
1050    if(rlt2.eq.0) go to 1070
        if(.not.lnav) go to 820
c ---- WRITES TIME/POSITION OF 2ND CDI RECORD TO WORKING DIRECTORY IN BINARY
c       FORMAT - PROVIDED NAV RECORD'S TIME IS > TIME OF 2ND CDI RECORD ----
        write(12) dy2,hr2,rmn2,rlt2,rln2
       write(13,8000) dy2,hr2,rmn2,rlt2,rln2
c ---- WRITES A DUMMY TIME AND POSITION RECORD HAVING ZERO LAT/LON VALUES TO
c       WORKING DIRECTORY. ZERO LAT/LON VALUES CAUSE PLOT PEN TO LIFT UP ----
        write(12) dy2,hr2,rmn2,rltd,rlnd
       write(13,8000) dy2,hr2,rmn2,rltd,rlnd
c ---- READS TWO MORE CDI RECORDS ----
        go to 820
1070    if(.not.lnav) go to 820
c ---- WRITES OUT TIME VALUES FOR LAST WRITTEN NAV RECORD BUT WITH NAV VALUES
c       = TO ZERO TO SIGNAL TO PLOTTER PEN TO LIFT OFF THE PAPER ----
        write(12) nvdh,nvhh,rmnh,rltd,rlnd
       write(13,8000) nvdh,nvhh,rmnh,rltd,rlnd
        go to 820
c ---- WRITES TIME AND POSITION OF NAV RECORD TO WORKING DIRECTORY IN
c       BINARY - PROVIDED NAV RECORD'S TIME FALLS BETWEEN THE TIMES FOR THE
c       TWO CURRENT CDI RECORDS ----
1100    lnav=.true.
        write(12) nvd,nvh,rmn,rltn,rlnn
       write(13,8000) nvd,nvh,rmn,rltn,rlnn
c ---- lmcdi = true => CURRENT NAV RECORD TIME IS BETWEEN THE TIMES FOR THE
c       TWO CURRENT CDI RECORDS ----
        lmcdi=.true.
c ---- STORES TIME VARIABLES FOR LAST WRITTEN NAV RECORD IN HOLDING
c       VARIABLES, nvdh,nvhh,rmnh ----
        nvdh=nvd
        nvhh=nvh
        rmnh=rmn
c
c ---- READS ANOTHER NAVIGATION RECORD ----
        go to (600,650,700), nb
c
c ---- SIGNALS END OF VIRTUAL FILE CONTAINING INPUT CDI ----
1500    write(6,1520) ncrin,ncdi
1520    format("0END OF CDI FILE ON UNIT 08.  NUMBER OF RECORDS READ = "i4,
              " FROM FILE NUMBER  ",i3)
c ---- nctot = TOTAL CDI RECORDS READ AT THIS POINT ----
              nctot=nctot+ncrin
              ncrin=0
              lbcdi=.false.
              lmcdi=.false.
c ---- WRITES NEGATIVE MINUTE VALUE TO OUTPUT FILE WHICH SIGNALS END-OF-CRUISE
c          TO PLOT PROGRAM AND INITIATES A NEW TRACKLINE PLOTTING SYMBOL ----
              write(12) nvd,nvh,rmnd,rltd,rlnd
              write(13,8000) nvd,nvh,rmnd,rltd,rlnd
c ---- ncdi ≥ TO ncf => LAST CDI RECORD HAS BEEN READ ----
1650          if(ncdi.ge.ncf) go to 5000
              if(nnav.ge.nnf) go to 1900
              go to 570
c
c ---- SIGNALS END OF A NAV FILE ----
1700          write(6,1710) nnrin,nnav
1710          format("0END OF NAV FILE ON UNIT 10 NUMBER OF RECORDS READ ="%
              ,i6," FROM FILE NUMBER  ",i3)
c ---- nntot = TOTAL NAV RECORDS READ AT THIS POINT ----
              nntot=nntot+nnrin
              nnrin=0
              lbcdi=.false.
              lmcdi=.false.
c ---- WRITES NEGATIVE MINUTE VALUE.  USES NEW PLOT SYMBOL ----
              write(12) nvd,nvh,rmnd,rltd,rlnd
              write(13,8000) nvd,nvh,rmnd,rltd,rlnd
c ---- nnav ≥ to nnf => LAST NAV RECORD IN LAST NAV FILE ON A GIVEN DEVICE
c          HAS BEEN ENCOUNTERED ----
1850          if(nnav.ge.nnf) go to 1900
c ---- BRANCHES TO MULTI_FILE STATEMENT WHICH SET UP CONDITIONS TO READ
c          NEW NAV AND CDI FILES ----
              go to 570
c
c ---- CLOSES NAV SWITCH ----
1900          ioeror=.true.
2200          call multi_file("file10",nnav,ioeror)
              if(ioeror) go to 6000
c
c ---- ASKS WHETHER THERE ARE FURTHER NAV FILES ON DIFFERENT DEVICES TO BE
c          ACCESSED ----
              write(6,2600)
2600          format(" ??ARE THERE MORE NAV FILES ON DIFFERENT STORAGE DEVICES "%
              "TO BE PROCESSED AT THIS TIME?? (t or f)")
              read(5,3000) lfnav
3000          format(a1)
              if(.not.lfnav) go to 6100
c
c ---- RE-INITIALIZES SWITCHES FOR NEW NAV STORAGE DEVICE ----
              nnav=0
              kn=kn+1
              write(6,150) ordnl(kn)
              read(5,100) nnf
              call mf_init(ioeror)
              if(ioeror) go to 4000
              go to 250
c
c ---- ENDS JOB AND PRINTS OUT TOTAL NUMBER OF CDI AND NAV RECORDS READ ----
4000          write(6,4100)
4100          format(" JOB TERMINATING DUE TO I/O ERROR.")
```

```
5000        nntot=nntot+nnrin
            write(6,5100) nctot,nntot
5100        format("0END OF JOB"/" TOTAL CDI RECORDS READ = ",i4/
            " TOTAL NAV RECORDS READ = ",i6)
c ---- CLOSES CDI AND NAV SWITCHES AT JOB TERMINATION ----
            ioeror=.true.
            call multi_file("file08",ncdi,ioeror)
            ioeror=.true.
            call multi_file("file10",nnav,ioeror)
            go to 6600
6000        write(6,4100)
6100        nctot=nctot+ncrin
            write(6,5100) nctot,nntot
c ---- CLOSES AND DETACHES OUTPUT SWITCH ----
6600        call io ("close","file12")
            call io ("detach","file12")
            stop
            end
```