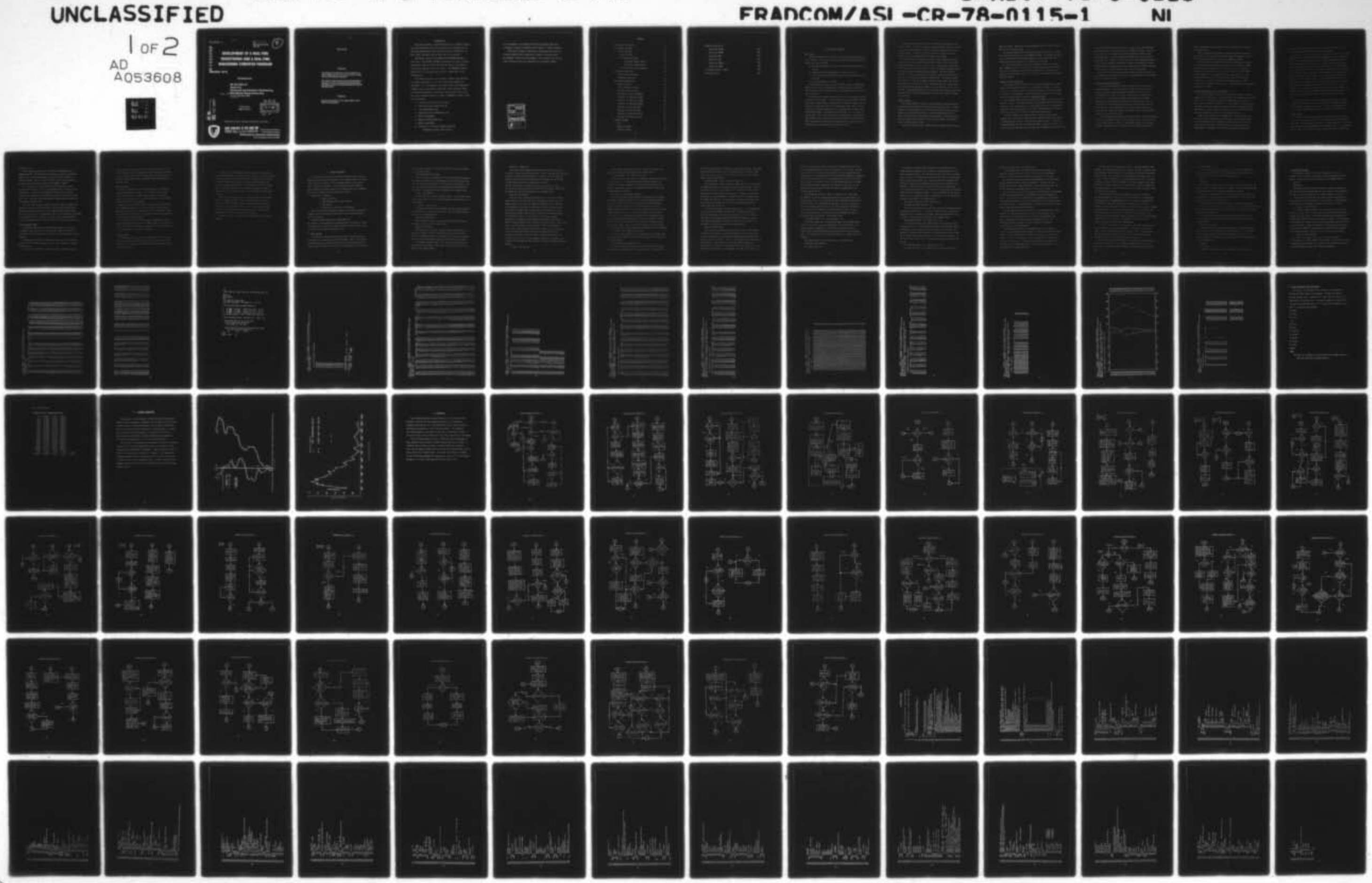
AD-A053 608 NEW MEXICO STATE UNIV LAS CRUCES F/G 4/2
DEVELOPMENT OF A REAL-TIME ROCKETSONDE AND A REAL-TIME RADIOSON--ETC(U)
JAN 78 M D MERRILL, S FRY DAAD07-76-C-0115
UNCLASSIFIED FRADCOM/ASL-CR-78-0115-1 NL

1 OF 2
AD
A053608



ASL-CR-78-0115 - 1

AD

Reports Control Symbol
OSD-1366

DEVELOPMENT OF A REAL-TIME ROCKETSONDE AND A REAL-TIME RADIOSONDE COMPUTER PROGRAM

JANUARY 1978

Prepared by

M. Don Merrill
Scott Fry
Electrical and Computer Engineering
New Mexico State University
Las Cruces, New Mexico 88003

Under Contract
DAAD07-76-C-0115

US Army Electronics Research and Development Command
Atmospheric Sciences Laboratory
White Sands Missile Range, N.M. 88002

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

INTRODUCTION

This report contains a detailed description of a computer program that was developed for use with the Interdata 7/32 computer and the interfaced Nike Hercules radar systems located at the MTTR site at White Sands, New Mexico and the Poker Flat site in Alaska.

The program contains both assembler and FORTRAN language instructions. The assembler language instructions are used to provide input/output control for data flow from the radar to the computer and from the computer to the T.V. display. The FORTRAN statements comprise 95% of the program and are used to compute most of the processed data.

The program can process in real-time a complete data reduction for an MRN rocketsonde or a list of 2 minute layer winds for a radiosonde. For a rocketsonde, the program utilizes the temperature telemetry data, the positional radar data, and the operator inputs of rocketsonde temperature calibration values and base level tie-in data from a radiosonde flight to produce an MRN(WDC-A) format listing of the following:

1. 1 KM corrected and uncorrected winds
2. Significant level temperature data
3. 1 KM thermodynamic data
4. Significant level thermodynamic data
5. MRN 30 cards(image)
6. Mandatory thermodynamic data
7. MRN 40 cards(image)
8. Printer plot of X and Y component winds and temperatures versus 1 KM altitudes

For a radiosonde, the program utilizes the positional radar data to produce a listing of two-minute layer winds at 1 minute intervals.

This report includes a description, listing and flow charts for the main program and all subroutines, together with instructions and examples on how to use the program. Also included is the typical output listing for both the radiosonde and rocketsonde flights.

CONTENTS

Program Listings(cont)

I. DESCRIPTION OF PROGRAMS

Main Program

Besides performing calls to the subroutine, the main program performs many other functions. At the start of execution the main program obtains necessary information from the operator:

1) what mode the program is to run in; real-time, calculation with tape, or copy;
2) whether a real time flight might be a rocket or a balloon;
3) for a real-time flight information such as name, time of launch, and date; and
4) if calibration data is available.

The main program contains an area of variables, found in the COMMON area BUFDTA, that is used as a storage buffer. The entire buffer is written in a write to magnetic tape on a real-time run, or read from mag tape on a cal with tape or copy run.

The main program contains the routine which, for a real-time run, services an interrupt from the ranger. This routine, starting at label INTRP, reads the data lines to obtain current, raw data values for elevation, azimuth, and slant range for the radar and to get the time and telemetry (met) values. The met values are run through a selective filter to determine bad, or noise, values, and determine changes or modulations in the met signal. For more detail on the met filter refer to the flow chart for the main program. Starting at the label DISPINT is the routine which services an interrupt from the TV display. This routine takes the latest raw data values and writes the data out to the display memory to be displayed on the next sweep of the TV.

The label RDTAPE begins an area of the main program which manipulates the raw data stored in the buffer. Elevation, azimuth, and range values are converted to a special integer format to be used by the digital smoothing filter in the subroutine WNDCAL. Time, which in raw data form has to be stored as three separate words, is converted to time, in seconds, after launch.

In the copy mode no subroutines are called, and the program merely reads a record from the mag tape and prints all levels of data in that record, reads another record, and so on until an end of file is encountered on the mag tape. For real-time and call with tape modes, the heart of the main processing routine starts at the FORTRAN lable 23Ø in the main program. The routine consists basically of shifting raw data values of elevation, azimuth, range, and time in and out of a 121-point data storage array labelled RAW. The subroutine WNDCAL is then called to filter the raw data, compute and correct winds, and to print layer data on the line printer. In the cal with tape mode, new raw data values are read from the mag tape until an end of file is encountered. For the real time mode, as previously mentioned, new raw data values are obtained in the range interrupt service routine; continuing until an end of flight signal is entered by the operator.

Subroutine WNDCAL

The subroutine WNDCAL is called by the main program in both real-time and calculation with tape modes, and for both rocket and balloon flights. The main purposes of the subroutine are: to smooth raw data values by using a 117-point, symmetrical, low-pass digital filter; to calculate positional data and to compute wind velocities; to apply wind correction methods to the wind velocities for rocket flights; and to print layer data on the line printer at the proper time or altitude during the flight.

The raw data values are sent to WNDCAL in the array IRAW, which is found in

COMMON area BLOCK1. BLOCK1 also contains parameters which are control variables and are set in the main program.

The subroutine WNDCAL stores computed and corrected layer data in a storage area in the main buffer, which is the COMMON area BUFDTA. The storage area starts at the variable named RTT and extends through the variable named RVS. This is the data that will be available to the program in the copy mode.

For rocket flights WNDCAL also stores the computed layer data in another area - COMMON area BLOCK2. The data saved consists of wind speed and direction, fall velocity, altitude, and time for each even kilometer layer during the descent of the rocketsonde. This data is later used in the temperature processing subroutine TEMCOR.

Subroutine SIGLEV (I1SIG, D55)

The subroutine SIGLEV is called only by the main program and only for rocket flights with temperature processing. The main purpose of SIGLEV is to detect, by the use of a selective filter, all significant changes in the met data during the descent of the rocketsonde. Time and temperature are saved for each change detected, and these points are marked as significant levels.

The subroutine SIGLEV performs different jobs depending on which of the three modes it is called. The passing parameter I1SIG determines what the mode will be, and is set by the calling program. The three modes of calls are: initialize and set up the routine the first time it is called; a normal temperature processing call; and the last call to print out all saved significant levels.

For normal temperature processing SIGLEV is called everytime the main program's met filter passes a "good" data point. The met ordinate value and its corresponding time are sent to SIGLEV via user registers 6 and 7. SIGLEV first determines whether the value is a regular temperature value or a reference. All references and the time they occur are saved, and all temperatures (and their times) that are distinguished as significant levels are also saved. The significant level temperatures and times are stored in the COMMON area LKTMTP, so that they may be used later by the temperature correction subroutine TEMCOR. SIGLEV has to make a call to subroutine LOKTEM to determine the actual uncorrected temperature value in degrees. SIGLEV sends a ratio of the temperature ordinate to its reference ordinate to LOKTEM, which returns the temperature in degrees. LOKTEM contains the calibration values for the particular rocketsonde instrument. For more detail on SIGLEV's selective filter refer to the flowchart.

On the last call to SIGLEV, the routine prints out the entire stored list of significant level temperatures and their corresponding times, and the complete list of references and times.

Subroutine LOKTEM (ICAL, CRATIO)

The subroutine LOKTEM has two primary functions and is called only for rocketsonde flights. For its first function, LOKTEM is called by the main program to determine whether temperature processing will be done or not. If temperature processing is to be done, and the program is running in the real-time mode, then calibration values for this particular rocketsonde instrument must be read in from a paper tape. If the program is in the cal with tape mode, then LOKTEM checks to see if the calibration data was saved on the mag tape; if not, LOKTEM asks the operator to input a paper tape. In either mode, if a paper tape is input, the operator is asked to make a simple check for verification of the data.

Normal temperature processing is the second function the subroutine performs. This occurs when LOKTEM is called by the subroutine SIGLEV. LOKTEM is passed a value, CRATIO, which is a ratio of a temperature ordinate to its reference ordinate. LOKTEM looks into its table of stored ratios and interpolates to obtain a temperature value in degrees, which is then returned to SIGLEV via the variable CRATIO.

ICAL is the name of the parameter that signals LOKTEM which function it is to perform. The variable that determines whether the program is in real-time or cal with tape mode is JC, which is in COMMON area BLOCK1.

When the program is running in real-time and a calibration tape is read in, all calibration data is saved by storing it in the main buffer, COMMON area BUFDTA, which is written on the first mag tape record. Appropriate flags are set, also in the buffer, to flag that the data is there. LOKTEM may then determine if the calibration data was saved so that it can read the data from the buffer in the first mag tape record. For either mode, if no cal data is available, LOKTEM sets the parameter ICAL to a value to signal the main program that no cal data is available and that no temperature processing can be done.

Subroutine TEMCOR (NODF, BLALT, BLPR, BLTMP)

The subroutine TEMCOR is called only once during a flight, by the main program, and is called only for rocketsonde flights where temperature processing is to be done. The routine is called in both real-time and cal with tape modes. The main purposes of TEMCOR are: to take temperatures computed during the flight and apply the standard IRIG correction method to obtain corrected temperatures; to compute pressures and densities, if an acceptable base level matchup can be made with a conjunctive rawinsonde observation.

TEMCOR is sent input data in two COMMON areas - BLOCK2 and LKTMTP. The first area contains the wind data which was stored by the subroutine WNDCAL at each even kilometer level. COMMON area LKTMTP contains all significant level temperatures and times, which were stored by the subroutine SIGLEV. TEMCOR obtains a temperature for each kilometer through interpolation, by time, into the significant temperature array; and finally a correction term is calculated which is added to the uncorrected temperature.

For the base level tie in, data must be input by the operator. BLALT, BLPR, and BLTMP are the input variables which are, respectively, base level altitude, pressure, and temperature. There are two criteria which determine whether particular base level data is good or not. First, it must be at an altitude reached during the descent of the rocketsonde while it was still transmitting met data; and second, the base level temperature must agree within 2.5 degrees of that measured by the rocketsonde at the same altitude. If a base level is rejected the operator may try another, and so on, until either a matchup is found or the operator signals to stop. In the case where no base level is found, obviously no pressure data can be computed, and a flag is set to signal the remainder of the program that no pressure data is available. If the program is in the real-time mode and a base level is accepted by TEMCOR, this data is written into the buffer, COMMON area BUFDTA, along with flags, so that the base level matchup will be saved on the last buffer write to the mag tape.

Corrected temperature, correction amount, pressure, density, and speed of sound are all saved in the remaining storage areas of COMMON area BLOCK2; this area then will contain the complete set of wind and thermodynamic data for 1-KM levels. TEMCOR prints this same set of data for each 1-KM level.

Subroutine SLML

The subroutine SLML performs a number of functions; but is only called once; by the main program, and only for rocketsonde flights with temperature processing. The main purposes of SLML are: to compute and print out thermodynamic data for significant and mandatory (constant pressure) levels; to call subroutine MRN for each computed layer of data; and to encode the data in the MRN format and print it out.

SLML is sent the complete array of 1-KM level thermodynamic data in COMMON area BLOCK2. The wind and positional data was computed by subroutine WNDCAL, and the corrected temperature and pressure data was computed by subroutine TEMCOR. SLML is also sent the array of significant temperatures and times, saved by subroutine SIGLEV, in the COMMON area LKTMTP.

First, SLML takes each significant level and interpolates by time, into the 1-KM level array. Thus, all thermodynamic data is obtained for each altitude where a significant level temperature was detected. Next SLML interleaves, by altitudes, this significant level array of data with the 1-KM level array. SLML then calls subroutine MRN to format and print out the data. This comprises the MRN 30 cards.

If no base level was given, mandatory level data cannot be computed, and execution is returned to the main program. Otherwise SLML determines which mandatory (constant pressure) levels fit between the upper and lower limits of the 1-KM level array. All thermodynamic data is then interpolated, by pressure, for each mandatory level. This data is printed, and then subroutine MRN is called to format and print out the data for each level (making up the 40 cards).

Subroutine MRN(ID, MRNA)

The subroutine MRN generates an MRN-formatted message for one level of thermodynamic data, when called by subroutine SLML. MRN is called only for rocketsonde flights with temperature processing, for both real-time and cal with tape modes.

The parameter ID identifies for MRN what type of data is to be formatted: 30 cards are 1-KM or significant levels, and 40 cards are mandatory (constant pressure) levels.

The data is sent to MRN in the passed array MRNA. Essentially the same format is used for both 30 and 40 cards, with two exceptions. For the 40 cards (mandatory levels), geopotential rather than geometric altitude is used, and fall velocity is not included. MRN accepts the passed data, formats it, prints out one line (a card image), and returns control to subroutine SLML.

Subroutine XYPLOT

The subroutine XYPLOT is called by the main program for rocketsonde flight with temperature processing. It is called for both real-time and cal with tape modes. The main purpose of XYPLOT is to construct a plot, on the line printer, of the winds and temperature versus altitude during the descent of the rocketsonde.

XYPLOT is sent the complete array of 1-KM level wind and thermodynamic data in COMMON area BLOCK2. For each 1-KM altitude the routine merely determines the values of the component winds and the temperature. XYPLOT then prints out one line giving the altitude, and marks in the appropriate spots for the winds and temperature.

An X stands for the value of the corrected X-component wind, and a Y stands for the value of the corrected Y-component wind. A T stands for the value of the corrected temperature. Anytime two or more of the parameters occupy the same spot on the plot an 0 is printed.

Subroutine ROCOB

The subroutine ROCOB is called by the main program for a rocketsonde flight with temperature processing, and for both real-time and cal with tape modes. It is called to perform the task of formulating and printing out the ROCOB message.

Formulation of the ROCOB message consists basically of running checks on the three parameters - wind speed, wind direction, and corrected temperature to detect layers which exceed specified limits (flowchart provides details on limits). These layers, plus those at every even 5KM throughout the flight, are included in the message. Thermodynamic data for each included layer is specially formatted and the parameter which exceeded limits and caused a layer to be selected is marked; this comprises the ROCOB message printout.

ROCOB is sent the complete array of 1-KM level wind and thermodynamic data in the COMMON area BLOCK2. Pressure and density are not used as parameters in making the data checks, so a ROCOB message is made whether a base level matchup was achieved or not. If it was, then density data is also formatted and included in the message printout; otherwise dummy values (9's) are filled in for density in the printout.

A complete listing of all programs is contained in the section labeled PROGRAMS.

II. OPERATING PROCEDURE

The program was written for operation on an Interdata model 7/32 minicomputer and to execute in the Interdata 32-bit operating system. This system requires that any I/O devices used by a program be given certain logical unit, or numerical, assignments. After the program has been loaded into memory, and before starting execution, the following logical unit assignments must be made to comply with those needed by the program:

Logical Unit - I/O Device

1 - Magnetic tape drive

2 - High speed paper tape reader and punch

3 - Line printer

5 - Command console (teletype or CRT terminal)

When the start execution command is entered by the operator, through the command console, the program will always write out the following message back to the console:

REAL TIME CR PRINT COPY 1CR CAL WITH MAG TAPE 2 CR.

The operator's response determines the running mode for the program. A step-by-step procedure is given here to explain available options for each of the three modes; and following are examples of each.

A. REAL TIME MODE

The program is placed in the real time mode by a CR (carriage return) in response to the initial question from the program. NOTE: If the operator anticipates running in real time, then prior to starting program execution a mag tape should be loaded on the tape drive and positioned to the proper point.

The first program action, in the real time mode, is to write this message to the command console:

INPUT RNAME'RDNM'DY'MN'YR'LNTM'

This is a request to the operator for the following flight information: round name (radiosonde, etc.), round number, day of the month, month, year (last two digits) and launch time. Data is entered by the operator on the next line, directly underneath the corresponding heading and between the quote marks. The program now writes this message to the command console:

ROCKET? 1 CR, SONDE? CR

Here the operator enters the appropriate answer - a CR (carriage return) for a balloon flight, or a 1 CR (a one followed by a carriage return) for a rocket flight.

1. Balloon - real time mode

The program will now print a page heading, with the flight information, on the line printer, and then a line heading. Next, the following message is written to the command console:

PRESS CR TO START DISPLAY

When the program receives a CR (carriage return) from the operator, this message is written to the command console:

PRESS DTA 2 AT LAUNCH

At the same time, the external interrupts from the ranger and the T.V. display are enabled. Raw data is now being read on the data receiving lines, by the program, and values are encoded to be written to the display; no data is yet being recorded on mag tape or written on the line printer.

At the moment of balloon lift-off, the operator should press the DTA and 2 buttons on the computer control panel. The program then writes this message to the command console:

PRESS DTA 3 AT RADAR LOCK

At the same time the program reads the time from the clock and writes out the Zulu launch time on the line printer. Also, at the same time, the program now starts recording data on the mag tape. Wind data is not yet being filtered or printed out, but raw positional data is being stored, and met data is being filtered and stored in the mag tape data buffer.

When the tracking radar system has confirmed radar lock on the target, the operator should press the DTA and 3 buttons on the computer control panel. The program will then write this message to the command console:

PRESS DTA 8 TO END FLIGHT

At the same time, the program prints the time, in minutes and seconds, between balloon lift-off and radar lock. The program now enters its fundamental running cycle. The cycle consists of these basic steps: when the ranger generates an interrupt, the data lines are read to accept new values for elevation, azimuth, range, and time; when the data buffer fills, it is written on the mag tape; raw data values are written to the display memory, to be displayed by the T.V.; data values are filtered and positional and wind data is computed; at every multiple of 60 seconds after balloon lift-off, wind and positional data is written to the line printer. This program cycle will continue until the operator enters a 'DTA 8' on the computer control panel. The program will then continue reading new data long enough to fill the data buffer; then a last write is issued to the mag tape, program execution ends, and control returns to the operating system. The operator should then issue a 'write filemark' command to the mag tape. This will allow for recording more than one flight on a tape, and keeping the flights separate.

2. Rocket - real time mode

When the opeator has indicated that the flight will be a rocket, the program will respond with this message to the command console:

CAL TAPE AVAILABLE? YES PRESS CR, NO 1 CR

The question really asked here is if temperature processing is to be done for this flight. If so, a paper tape, with calibration values for the particular rocketsonde instrument to be tracked, must be read in by the program. Instructions on making a calibration paper tape are given in the following section of examples.

2a. Rocket - no temperature processing

For a real-time rocket flight, where no calibration data is available, and thus no temperature processing is to be done, the operating procedure is the very same as described in the previous section (under heading: 1. Balloon - real time mode), with a few minor exceptions. First, the operator should enter 'DTA 2' on the computer control panel at rocket launch rather than balloon lift-off. The other exceptions are in the program cycle itself. For a rocketsonde track, wind data is calculated and also corrected using correction techniques for vertical acceleration. Finally, instead of data being printed on the line printer at 60 second intervals after balloon lift-off, now corrected winds and angular data are printed out for every 1-KM layer passed through during the descent of the rocketsonde.

2b. Rocket - with temperature processing

When the operator indicates that a calibration tape is available (tape must be prepared in advance), the program will respond with this message on the command console:

PUT LOKI TEMP TAPE IN READER PRESS CR

The operator should then load the cal tape in the paper tape reader, and then hit the carriage return on the command console. The tape will be read in, and the program will construct a table of look-up values. The program then takes three different ratios and determines their corresponding values in the look-up table. The ratios and values are printed out on the command console, and then this message:

CAL RATIOS WITHIN .3 DEG? PRESS CR, NO PRESS 1 CR

Here the operator is asked to perform a simple check to see if the values calculated and stored by the program fall within acceptable limits Again for more details here, refer to the example. If the ratios do not agree, then temperature processing cannot be done correctly. In this case, the operator should type 1 CR (one followed by a carriage return) on the command console: the program will then stop. The operator may try constructing another tape, in case the first had errors, or he may start the program again and run without temperature processing.

If the ratios are okay the operator should enter a CR. The program will list out on the line printer all ratios read in and the coefficients calculated by the program. The line printer then goes to the top of the next page, prints a page heading (again with the flight information), prints a line heading, the program is now ready for the start of the flight. The program will now write this message to the command console:

PRESS CR TO START DISPLAY

From this point in the program, up to where the 'end of flight' signal is entered by the operator, the operating procedure is the very same as for the real time balloon, or rocket without temperature processing (described in the two previous sections). There are a few exceptions in the program cycle itself though. As the program does for a rocket without temperature processing, so here winds are corrected, and data is now printed out for each 1-KM layer in the descent of the rocketsonde. Also here, as met values are being read, filtered, and stored, some special processing will be done. First, it is determined whether a met value is a temperature or a reference. All temperature values are run through a selective filter to detect significant level temperatures; for each one detected an actual uncorrected temperature, in degrees, will be computed by making use of the calibration values read in at the start of the program. All reference values, and their corresponding time, are saved, and all significant level temperatures, and their times, are saved. If the operator desires, these values may also be listed, by assigning some list device to logical unit 8.

When the operator enters the 'DTA 8' to signal the end of the flight, the program will continue reading and processing new data long enough for the data buffer to fill up. All interrupts will then be disabled, and no new data can be processed. The line printer will now move to the top of a new page, print a page heading, and then list out all stored temperature references and all significant level temperatures detected during the flight. The line printer will again move to a new page, print a page heading; and then the program will write this message to the command console:

DAY OR NIGHT FLIGHT? Day - Press CR; NIGHT - 1 CR

The operator should enter the appropriate answer; this determines which values will be used in correcting temperatures for incoming radiation. The program computes a temperature for each 1-KM layer in the wind array (if between 20 and 70 KM) by interpolating into the significant level temperature array. The temperatures are then corrected using the U.S. standard correction technique.

The program will then write this message to the command console:

IS BASE LEVEL DATA AVAILABLE?

YES - PRESS CR; NO 1 CR

To be able to compute pressure and densities, the program must obtain a suitable base level match-up from a conjunctive rawinsonde observation. Two criteria determine whether or not some particular base level data is suitable. First, it must be at an altitude reached during the descent of the rocketsonde while it was transmitting met data. Second, the base level temperature must be within 2.5 degrees of the temperature measured by the rocketsonde at that altitude. If no base level data is available, the operator should enter a 1 CR; the program then will fill in dummy values for all pressures and densities in the 1-KM level thermodynamic array. If data is available, and the operator wants to try it as a possible matchup, he should enter a CR. The program will then write this message to the command console:

TYPE IN BASE LEVEL DATA (INCLUDE DECIMAL POINT)

'ALT (KM)' PRSS (MB)' TEMP (DEG - C)'

The operator should then type in the data, under the corresponding headings and between the quote marks. If the data does not pass the two criteria the program will write a message to the command console explaining which of the two failed. Then this message is written to the command console:

TRY ANOTHER BASE LEVEL? YES - PRESS CR; NO - 1 CR

If the operator desires to try different data for a possible match-up, he should enter a CR. The program will then write the above message (asking the operator to type in base level data), and the cycle is repeated. The operator may continue trying to achieve an acceptable match-up, until either one is achieved or the operator decides to stop trying for the match-up. If the operator does decide to stop, a 1 CR should be entered in response to the question:

TRY ANOTHER BASE LEVEL? YES - PRESS CR: NO - 1CR

The program will then fill in dummy values for all the pressures and densities in the 1-KM level thermodynamic array.

If an acceptable base level match-up is entered by the operator, the program will compute a pressure for each 1-KM level using the hydrostatic equation. Density and speed of sound are also computed for each level.

The program will now print the base level information on the line printer, and then the entire 1-KM level array is printed out. The print out will include altitude, time, corrected wind data, corrected temperature data, and pressure and density (with dummy values if no base level is found).

The line printer will now go to the top of a new page, print a page heading, and then the program will print out thermodynamic data for all significant levels. The data print out is the same form as described above for the 1-KM level thermodynamic array. The data is calculated by direct interpolation into the 1-KM level array. A significant level is created at each time in the flight where a significant temperature was detected.

The line printer will again go to the top of a new page and print a page heading. The program will now print out data in the special MRN format - the MRN 30 cards. Each line consists of one level of thermodynamic data. The 30 cards include all 1-KM levels and all significant levels. The levels are arranged by altitude, with each significant level between the two 1-KM levels above and below it.

The program will now print out thermodynamic data for all mandatory levels (on a new page with a page heading). A mandatory (or constant pressure) level is one where the pressure, in millibars, is equal to any powers of 10 times 7, 5, 3, 2, or 1. This data is also calculated by direct interpolation into the 1-KM level array. The data is next printed on a new page in the MRN format. These are the MRN 40 cards, each line of data consists of one mandatory level of thermodynamic data.

The line printer will again go to the top of a new page and print a page heading. The program will now print out a graph of the X and Y component winds and temperature, versus altitude. The vertical axis is altitude in kilometers, above sea level. Three parameters are plotted along the horizontal axis - X, Y, and T. The X's represent the corrected component wind in the X - direction, in meters per second. The Y's represent the corrected component wind in the Y - direction, in meters per second. The T's stand for the corrected temperature, in degrees centigrade. Any O's means that two, or all three, parameters occupy the same point on the graph.

The program will now print a page heading at the top of a new page. The ROCOB message will now be printed on the line printer. The ROCOB message consists basically of thermodynamic data for all levels with altitudes that are multiples of 5KM. Also included are all levels where one of three parameters - wind speed, wind direction, or temperature - deviate from a straight line approximation by more than a specified amount. The print-out gives the altitude of each level of data and the value of all three parameters. An X will appear in the appropriate column to indicate which of the three parameters caused that level to be selected. The numbers at the far right of the print-out are the data in the special ROCOB format.

A last write is now issued to the mag tape. This record will contain the base level values if a match-up was achieved. Also on this record will be flags to indicate whether base level data is there or not. If the flags are set, and the flight is run at a later time in the cal with tape mode, the program will know to use the base level values already there.

Program execution now ends and control is returned to the operating system. The operator should issue a 'write filemark' command to the mag tape. This will allow for keeping this flight separate from the next one recorded on the tape.

B. PRINT COPY MODE

The program is placed in the copy mode by a 1 CR (a one followed by a carriage return) in response to the initial question from the program. The program will return this message to the command console:

PAUSE 10

TASK PAUSED

The program is paused to allow the operator to position a mag tape to the flight (previously recorded) which is to be copied. After positioning the tape, the operator should enter a 'continue' command on the console.

The program will now read the first record of the mag tape and print a page heading with flight information on the line printer. Flight launch time and radar lock time are also printed out.

Following this, the program will read the mag tape and create a print-out of data. The print-out will include filtered and calculated wind and positional data. This data was saved during the real time run, and this mode merely copies the data from the mag tape.

The program will continue reading data from the tape until an 'end of file' mark is encountered, at which time this message will be written to the command console:

DONE? YES CR, NO 1 CR

If the operator desires to end the program here, a CR should be entered and program execution is returned to the operating system. If it is desired to run the same flight again or another flight in the copy mode, a 1 CR should be entered by the operator. The program will then return this message to the command console:

PAUSE 10

TASK PAUSED.

Operation is exactly the same as described above for the first copy.

C. <u>CAL WITH MAG TAPE MODE</u>

The program is placed in the cal with tape mode by a 2 CR (a two followed by a carriage return) in response to the initial question from the program. The program will return this message to the command console:

PAUSE 10

TASK PAUSED

The program is paused to allow the operator to position a mag tape to the flight (previously recorded) which is to be processed. After positioning the tape, the operator should enter a 'continue' command on the console.

The program will now read the first record of the mag tape and print a page heading with flight information on the line printer. Also determined in this first tape read is whether the flight was a balloon or a rocket.

1. Balloon - cal with tape mode

For a balloon flight the program will read the mag tape and create a print-out of data. First the flight launch time and the radar lock time are printed on the line printer. The program then reads raw data values from the mag tape, and filters the data and computes the winds. At every multiple of 60 seconds after balloon lift-off the wind and positional data are printed on the line printer. The program continues to read the tape until an 'end of file' mark is encountered. The program will stop and control is returned to the operating system. See pages 21 and 22 for actual wind listings.

2. Rocket-cal with tape mode

On page 23 is a sample print out on the TTY and the response by the operator for a rocket flight. Also included on pages 24 thru 33 is the corresponding output produced by the computer. Note each time the character < occurs the operator must supply the appropriate responses.

RADIOSONDE NUMBER 83 LAUNCHED 28 SEP 76 520 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AURSS
GEOMETRIC ALTITUDES
ZULU LAUNCH TIME 5:18:49
RADAR LOCK TIME= 0: 0:26

TIME (SEC)	ALTITUDE(MSL) (KM)	ALTITUDE(MSL) (KFT)	WIND VELOCITY(M/SEC) -N+S	WIND VELOCITY(M/SEC) -E+W	WIND VELOCITY(M/SEC) TOTAL	DIR (DEG)	FL VEL (M/SC)	WND SHEAR (MPS/M)	TIME (MIN SEC)	POSITION +N-S	POSITION +E-W	ANGLES(DEG) EL	ANGLES(DEG) AZ
26.0	0.3	1.0	0.0	0.0	0.0	1300.6	0.	0.000	0.26	966.97	-881.73	-4.084	317.640
26.0	0.3	1.0	-4.1	-9.8	10.6	67.4	5.	104.532	0.26	966.97	-881.73	2.304	301.184
60.0	0.5	1.6	-6.1	-13.6	14.9	65.7	4.	0.011	1.00	769.58	-1271.55	7.819	280.302
120.0	0.7	2.3	-4.4	-15.7	16.3	74.2	4.	0.005	2.00	392.05	-2157.01	10.065	274.290
180.0	1.0	3.3	-0.7	-15.2	15.2	87.5	5.	0.007	3.00	236.36	-3150.92	12.092	274.507
240.0	1.3	4.3	2.1	-13.0	13.2	99.1	5.	0.006	4.00	313.45	-3976.39	13.547	275.908
300.0	1.6	5.2	4.1	-12.5	13.2	108.0	5.	0.004	5.00	487.80	-4713.96	14.175	278.339
360.0	1.8	5.9	5.2	-12.5	13.5	112.5	4.	0.002	6.00	803.21	-5480.00	14.454	280.127
420.0	2.0	6.6	5.3	-12.5	13.6	113.0	4.	0.000	7.00	1109.60	-6212.41	14.533	281.648
480.0	2.3	7.5	4.5	-12.1	12.9	110.4	4.	0.002	8.00	1439.15	-6981.25	14.973	282.150
540.0	2.5	8.2	3.0	-11.9	12.3	104.1	4.	0.003	9.00	1650.37	-7665.91	15.165	282.066
600.0	2.8	9.2	3.6	-12.0	12.5	106.7	4.	0.001	10.00	1798.62	-8414.49	15.454	282.879
660.0	3.0	9.8	5.5	-11.6	12.8	115.5	4.	0.004	11.00	2082.43	-9108.09	15.728	284.088
720.0	3.3	10.8	8.1	-11.5	14.0	125.3	4.	0.005	12.00	2459.53	-9800.77	15.871	286.249
780.0	3.5	11.5	8.2	-9.4	12.5	131.1	5.	0.004	13.00	3055.05	-10482.29	16.451	287.487
840.0	3.8	12.5	6.5	-7.7	10.1	130.2	5.	0.004	14.00	3442.83	-10928.18	16.859	288.598
900.0	4.1	13.5	5.9	-7.4	9.4	128.6	5.	0.001	15.00	3839.30	-11409.84	17.424	289.348
960.0	4.4	14.4	5.8	-6.1	8.4	133.4	5.	0.002	16.00	4147.19	-11810.79	17.913	290.468
1020.0	4.6	15.1	7.1	-4.4	8.3	148.4	4.	0.004	17.00	4532.14	-12142.54	18.480	292.053
1080.0	4.9	16.1	10.0	-1.4	10.1	171.9	4.	0.008	18.00	4996.41	-12333.64	19.136	294.953
1140.0	5.1	16.7	12.0	0.4	12.0	182.1	5.	0.005	19.00	5728.72	-12311.93	19.850	297.655
1200.0	5.4	17.7	12.5	-1.5	12.6	173.0	5.	0.004	20.00	6435.76	-12281.89	20.096	300.041
1260.0	5.7	18.7	13.1	-3.6	13.6	164.5	5.	0.004	21.00	7225.78	-12494.76	20.315	302.193
1320.0	6.0	19.7	12.8	-3.6	13.3	164.4	5.	0.001	22.00	8006.61	-12717.88	20.466	304.135
1380.0	6.3	20.7	12.0	-3.6	12.5	163.1	4.	0.002	23.00	8762.12	-12924.41	20.561	305.671
1440.0	6.5	21.3	12.6	-4.2	13.3	161.7	4.	0.002	24.00	9441.63	-13153.30	20.525	307.428
1500.0	6.8	22.3	14.6	-3.5	15.0	166.5	4.	0.004	25.00	10273.08	-13423.06	20.533	309.503
1560.0	7.0	23.0	15.1	-2.3	15.3	171.5	5.	0.002	26.00	11190.34	-13573.69	20.617	311.426
1620.0	7.3	24.0	15.7	-1.1	15.8	176.2	5.	0.002	27.00	12083.80	-13693.89	20.684	313.667
1680.0	7.6	24.9	16.8	-1.1	16.9	176.1	5.	0.002	28.00	13077.08	-13700.32	20.652	315.559
1740.0	7.9	25.9	17.7	-2.6	17.9	171.7	5.	0.003	29.00	14103.44	-13831.11	20.514	317.341
1800.0	8.2	26.9	18.1	-2.6	18.3	171.9	5.	0.001	30.00	15205.12	-14010.70	20.353	319.022
1860.0	8.5	27.9	18.2	-2.5	18.4	172.3	4.	0.000	31.00	16280.20	-14141.41	20.165	320.555
1920.0	8.7	28.5	18.8	-2.4	18.9	172.7	5.	0.001	32.00	17388.53	-14305.92	19.982	322.097
1980.0	9.0	29.5	18.9	-1.7	19.0	174.9	5.	0.001	33.00	18532.87	-14429.27	19.838	323.569
2040.0	9.3	30.5	19.2	-1.4	19.2	175.8	5.	0.001	34.00	19655.01	-14507.64	19.673	324.981
2100.0	9.6	31.5	19.5	-1.9	19.6	174.3	5.	0.001	35.00	20836.05	-14600.11	19.495	326.175
2160.0	9.8	32.2	20.9	-2.6	21.0	173.0	5.	0.002	36.00	21998.16	-14740.77	19.299	327.436
2220.0	10.2	33.5	21.5	-1.5	21.5	176.0	6.	0.002	37.00	23342.75	-14907.84	19.303	328.736
2280.0	10.6	34.8	19.0	0.8	19.0	182.4	6.	0.005	38.00	24574.08	-14920.25	19.387	329.968
2340.0	10.9	35.8	19.3	0.3	19.3	180.8	6.	0.001	39.00	25619.16	-14818.70	19.315	331.034
2400.0	11.3	37.1	26.2	-1.3	26.2	177.2	6.	0.011	40.00	26895.13	-14887.63	18.846	332.513
2460.0	11.6	38.1	33.2	0.8	33.2	181.4	5.	0.012	41.00	28761.25	-14964.03	18.383	334.406
2520.0	11.9	39.0	34.6	4.1	34.9	186.8	6.	0.005	42.00	30877.65	-14789.83	18.090	336.266
2580.0	12.3	40.4	32.4	6.7	33.1	191.6	6.	0.005	43.00	32914.36	-14471.92	17.847	338.001
2640.0	12.6	41.3	28.5	5.8	29.1	191.6	6.	0.006	44.00	34764.55	-13988.88	17.686	339.239
2700.0	12.9	42.3	24.9	3.3	25.1	187.6	6.	0.006	45.00	36329.88	-13772.26	17.610	340.200
2760.0	13.3	43.6	23.9	1.3	23.9	183.1	6.	0.003	46.00	37748.95	-13590.26	17.532	340.844
2820.0	13.7	44.9	24.6	0.3	24.6	180.6	6.	0.002	47.00	39194.04	-13615.33	17.455	341.574
2880.0	14.1	46.3	22.4	1.9	22.5	184.8	6.	0.004	48.00	40695.41	-13558.46	17.455	342.271
2940.0	14.4	47.2	18.0	1.7	18.1	185.4	5.	0.007	49.00	41884.91	-13391.00	17.461	342.697
3000.0	14.7	48.2	15.1	-0.6	15.1	177.6	5.	0.006	50.00	42860.11	-13352.09	17.528	342.873
3060.0	15.0	49.2	14.7	-1.5	14.8	174.2	6.	0.001	51.00	43700.20	-13466.31	17.606	343.132

3120.0	15.4	50.5	13.5	0.3	13.5	181.5	6.	0.003	52.00	44621.77	-13530.41	17.772	343.501
3180.0	15.8	51.8	12.7	0.4	12.7	181.7	6.	0.001	53.00	45322.09	-13424.37	17.873	343.709
3240.0	16.1	52.8	13.1	0.1	13.1	180.4	6.	0.001	54.00	46142.52	-13484.88	17.985	344.037
3300.0	16.5	54.1	10.1	2.4	10.4	193.2	6.	0.005	55.00	46892.05	-13413.21	18.238	344.428
3360.0	16.8	55.1	7.3	1.5	7.4	191.5	6.	0.004	56.00	47360.10	-13198.61	18.485	344.510
3420.0	17.2	56.4	9.3	-0.7	9.3	176.0	6.	0.004	57.00	47762.08	-13236.63	18.648	344.683
3480.0	17.6	57.7	10.2	-0.5	10.3	177.4	6.	0.001	58.00	48476.54	-13277.09	18.843	344.820
3540.0	18.0	59.1	8.6	-0.2	8.6	179.0	6.	0.002	59.00	48991.71	-13292.50	18.985	344.969
3600.0	18.3	60.0	8.7	1.8	8.9	192.0	6.	0.003	60.00	49512.46	-13295.18	19.181	345.360
3660.0	18.6	61.0	5.2	3.3	6.2	212.0	7.	0.005	61.00	50036.21	-13070.85	19.577	345.569
3720.0	19.1	62.7	2.5	-0.1	2.5	177.4	7.	0.005	62.00	50140.34	-12903.38	19.857	345.429
3780.0	19.4	63.6	4.8	-2.2	5.2	155.6	6.	0.004	63.00	50334.21	-13084.19	20.048	345.449
3840.0	19.8	65.0	4.1	0.2	4.1	183.1	6.	0.003	64.00	50713.26	-13163.70	20.437	345.593
3900.0	20.2	66.3	2.4	-0.5	2.4	168.9	6.	0.003	65.00	50830.52	-13057.67	20.647	345.468
3960.0	20.5	67.3	4.6	-2.3	5.1	153.3	6.	0.004	66.00	51001.41	-13220.17	20.881	345.452
4020.0	20.9	68.6	5.2	1.0	5.3	190.9	7.	0.004	67.00	51380.74	-13333.99	21.221	345.760
4080.0	21.3	69.9	2.2	2.5	3.4	228.9	6.	0.005	68.00	51623.71	-13100.69	21.500	345.829
4140.0	21.6	70.9	2.1	2.4	3.2	228.8	6.	0.000	69.00	51645.27	-13030.75	21.816	346.124
4200.0	22.1	72.5	1.1	3.5	3.7	252.2	6.	0.002	70.00	51874.33	-12814.31	22.208	346.318
4260.0	22.4	73.5	-1.4	3.6	3.8	291.3	6.	0.003	71.00	51781.71	-12606.13	22.600	346.531
4320.0	22.8	74.8	-0.6	1.3	1.5	293.8	6.	0.003	72.00	51707.10	-12384.71	22.895	346.467
4380.0	23.1	75.8	1.0	-0.2	1.0	170.4	6.	0.003	73.00	51711.32	-12446.30	23.173	346.540
4440.0	23.5	77.1	1.7	1.7	2.4	224.3	6.	0.003	74.00	51829.16	-12405.38	23.477	346.728
4500.0	23.8	78.1	-0.1	2.9	2.9	272.6	6.	0.003	75.00	51916.56	-12245.87	23.879	346.903
4560.0	24.2	79.4	-1.5	0.8	1.7	332.1	6.	0.003	76.00	51813.25	-12054.28	24.224	346.785
4620.0	24.6	80.7	0.2	0.6	0.6	249.5	6.	0.002	77.00	51732.04	-12148.23	24.542	346.980
4680.0	25.0	82.0	0.7	2.9	2.9	257.1	6.	0.003	78.00	51838.55	-11986.73	24.851	347.165
4740.0	25.3	83.0	-2.2	3.1	3.8	305.6	0.	0.065	79.00	51811.06	-11804.55	24.723	347.306

```
*ST
  REAL TIME CR  PRINT COPY 1CR  CAL WITH MAG TAPE 2CR
>2
PAUSE 1Ø
TASK PAUSED
*CO
 CAL DATA NOT ON MAG TAPE
 CAL TAPE AVAILABLE?  YES PRESS CR ,  NO 1CR
>
 PUT LOKI TEMP TAPE IN READER PRESS CR
>
*    Ø.3ØØØ  -46.4958   (Temp from chart -46.34)
*    Ø.5ØØØ  -29.198Ø   (Temp from chart -29.27)
*    Ø.7ØØØ  -11.8978   (Temp from chart -11.92)
 CAL RATIOS WITHIN .3 DEG?  PRESS CR,NO PRESS 1CR
>
   DAY OR NIGHT FLIGHT?  DAY-PRESS CR;  NIGHT- 1CR
>1
   BASE LEVEL DATA IS NOT ON MAG TAPE
   IS BASE LEVEL DATA AVAILABLE?
     YES- PRESS CR;  NO- 1CR
>
   TYPE IN BASE LEVEL DATA (INCLUDE DECIMAL POINT)
     'ALT (KM)'PRSS(MB)'TEMP(DEG-C)'
>      28.       16.8     -5Ø.Ø
STOP
END OF TASK     Ø
*
```

LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AUASS
GEOMETRIC ALTITUDES

LOKI CALIBRATION THPE DATA

RATIOS
0.9910
0.9570
0.9180
0.8500
0.7930
0.7430
0.7000
0.6620
0.6290
0.5990
0.5730
0.5500
0.4590
0.3980
0.3210
0.2750
0.2220
0.1920
0.1740
0.1470
0.1330

CALIBRATION VALUES

1199.9067	430.0930	96.1588	26.9235	6.1105

QUADRATIC COEFFICIENTS

73.8988	-0.0254	0.0000
86.2863	-0.0309	0.0000

LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AVASS
GEOMETRIC ALTITUDES
ZULU LAUNCH TIME 7: 0: 0
RADAR LOCK TIME= 0: 2:11

TIME (SEC)	ALTITUDE(MSL) (KM)	ALTITUDE(MSL) (KFT)	WIND VELOCITY(M/SEC) -N+S	WIND VELOCITY(M/SEC) -E+W	WIND VELOCITY(M/SEC) TOTAL	DIR (DEG)	FL. VEL. (M/SC)	WND SHEAR (MPS/M)	TIME (MIN SEC)	UNC VEL -N+S	UNC VEL -E+W	ANGLES(DEG) EL	ANGLES(DEG) AZ
136.0	67.1	220.1	0.0	0.0	0.0	21214.1	0.	0.000	2.16	15329.57	14664.27	72.284	43.729
167.0	63.0	206.7	-16.4	28.8	33.2	299.7	-133.	0.035	2.47	-12.83	19.08	70.303	44.548
174.5	62.0	203.4	-6.3	28.7	29.4	282.4	-123.	0.010	2.54	-10.06	21.31	69.934	44.983
183.0	61.0	200.1	-6.2	23.3	24.1	284.8	-113.	0.005	3.03	-7.26	23.34	69.533	45.532
192.0	60.0	196.9	-4.4	21.9	22.3	281.3	-106.	0.002	3.12	-6.19	21.96	69.122	45.958
202.0	59.0	193.6	-2.3	22.6	22.7	275.8	-99.	0.002	3.22	-6.85	19.58	68.714	46.511
212.5	58.0	190.3	-2.8	25.0	25.1	276.5	-91.	0.002	3.32	-8.23	21.02	68.263	47.111
224.0	57.0	187.0	-0.6	19.1	19.1	271.9	-85.	0.006	3.44	-5.09	18.91	67.806	47.508
236.0	56.0	183.7	0.7	12.3	12.3	266.6	-81.	0.007	3.56	-1.65	13.51	67.348	47.769
249.0	55.0	180.4	-3.3	3.0	4.4	317.1	-76.	0.010	4.09	-2.23	5.83	66.981	47.889
263.0	54.0	177.2	-6.9	6.4	9.4	317.0	-71.	0.005	4.23	-5.66	5.39	66.578	48.322
277.0	53.0	173.9	-16.2	12.5	20.5	322.4	-67.	0.011	4.37	-13.67	10.12	66.228	49.177
293.0	52.0	170.6	-22.5	19.5	29.8	319.1	-63.	0.009	4.53	-21.26	17.05	65.781	50.527
309.0	51.0	167.3	-19.5	19.3	27.5	315.2	-59.	0.003	5.09	-21.01	19.03	65.366	51.545
326.0	50.0	164.0	-21.9	22.1	31.1	314.7	-56.	0.004	5.26	-22.15	21.05	64.821	53.216
344.0	49.0	160.8	-24.2	29.2	37.9	309.6	-53.	0.007	5.44	-24.45	26.45	64.203	54.934
364.0	48.0	157.5	-20.3	24.9	32.1	309.2	-50.	0.006	6.04	-21.39	25.43	63.547	56.431
384.0	47.0	154.2	-17.7	19.8	26.5	311.7	-46.	0.006	6.24	-18.91	20.66	62.939	57.842
407.0	46.0	150.9	-14.8	14.1	20.4	316.4	-43.	0.006	6.47	-15.32	14.61	62.334	58.843
431.0	45.0	147.6	-11.2	14.4	18.3	307.8	-39.	0.004	7.11	-11.48	14.48	61.557	60.009
458.0	44.0	144.4	-13.8	14.5	20.0	313.5	-36.	0.003	7.38	-13.41	14.00	60.922	61.392
487.0	43.0	141.1	-12.6	18.1	22.0	304.8	-33.	0.004	8.07	-12.60	17.59	59.815	62.870
519.0	42.0	137.8	-7.6	22.0	23.2	289.1	-30.	0.006	8.39	-8.44	21.84	58.659	64.139
553.0	41.0	134.5	-5.5	19.5	20.2	285.7	-28.	0.003	9.13	-5.77	19.54	57.453	65.143
590.0	40.0	131.2	-3.7	14.0	14.5	284.7	-26.	0.006	9.50	-3.71	14.33	56.437	65.798
630.0	39.0	128.0	-4.7	12.8	13.6	290.2	-24.	0.002	10.30	-4.61	12.63	55.261	66.884
673.0	38.0	124.7	-2.9	14.1	14.4	281.5	-22.	0.002	11.13	-2.94	13.92	53.942	67.411
719.0	37.0	121.4	-1.1	11.8	11.8	275.5	-21.	0.003	11.59	-1.34	12.07	52.755	68.065
766.0	36.0	118.1	-1.2	9.3	9.4	277.3	-20.	0.002	12.46	-1.32	9.39	51.562	68.412
818.0	35.0	114.8	-3.9	10.3	11.0	290.8	-18.	0.003	13.38	-4.03	10.18	50.207	69.834
880.0	34.0	111.5	-1.9	8.7	8.9	282.3	-16.	0.003	14.40	-1.94	8.76	48.913	69.705
945.0	33.0	108.3	-1.4	6.7	6.9	281.8	-15.	0.002	15.45	-1.33	6.66	47.699	70.757
1010.0	32.0	105.0	-3.2	9.2	9.7	289.2	-14.	0.003	16.50	-3.22	9.03	46.061	71.412
1087.0	31.0	101.7	-3.9	10.2	10.9	291.1	-13.	0.001	18.07	-3.92	10.12	44.536	72.873
1167.0	30.0	98.4	-1.9	8.6	8.8	282.2	-12.	0.003	19.27	-1.80	8.61	42.913	72.794
1253.0	29.0	95.1	0.6	5.9	5.9	263.7	-11.	0.004	20.53	0.67	5.88	41.514	73.245
1352.0	28.0	91.9	-0.4	7.4	7.4	272.9	-10.	0.002	22.32	-0.40	7.35	39.560	73.726
1457.0	27.0	88.6	2.6	10.2	10.5	255.5	-9.	0.004	24.17	2.60	10.17	37.422	73.409
1567.0	26.0	85.3	4.3	9.6	10.5	245.7	-9.	0.002	26.07	4.34	9.57	35.385	73.154
1688.0	25.0	82.0	5.5	9.2	10.7	239.3	-8.	0.001	28.08	5.44	9.16	33.184	72.374
1816.0	24.0	78.7	5.7	7.2	9.2	231.9	-7.	0.002	30.16	5.62	7.14	31.439	71.763
1957.0	23.0	75.5	6.4	5.8	8.6	222.5	-7.	0.002	32.37	6.35	5.80	29.390	70.473
2110.0	22.0	72.2	6.0	7.7	9.8	231.9	-6.	0.002	35.10	6.04	7.71	27.224	70.211
2281.0	21.0	68.9	8.2	6.9	10.7	220.1	-6.	0.002	38.01	8.21	6.87	25.208	67.879
2453.0	20.0	65.6	10.8	7.1	13.0	213.3	-5.	0.003	40.53	10.84	7.07	22.857	66.367

LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AVASS
GEOMETRIC ALTITUDES

SIGNIFICANT LEVEL PRINTOUT

TIME=	137	2:17	TEMP=	-36.1	0.4178	9371.1
TIME=	143	2:23	TEMP=	-39.4	0.3785	9373.0
TIME=	149	2:29	TEMP=	-39.5	0.3777	9374.9
TIME=	161	2:41	TEMP=	-32.8	0.4593	9378.7
TIME=	194	3:14	TEMP=	-31.2	0.4792	9389.1
TIME=	264	4:24	TEMP=	-11.3	0.7135	9411.2
TIME=	329	5:29	TEMP=	-9.8	0.7284	9431.7
TIME=	365	6: 5	TEMP=	-13.3	0.6935	9445.0
TIME=	389	6:29	TEMP=	-18.5	0.6365	9456.0
TIME=	437	7:17	TEMP=	-22.4	0.5909	9478.7
TIME=	473	7:53	TEMP=	-20.5	0.6125	9467.9
TIME=	503	8:23	TEMP=	-22.7	0.5863	9465.4
TIME=	533	8:53	TEMP=	-30.6	0.4880	9518.7
TIME=	564	9:24	TEMP=	-33.5	0.4501	9549.5
TIME=	636	10:36	TEMP=	-29.5	0.5018	9594.7
TIME=	685	11:25	TEMP=	-34.4	0.4396	9628.0
TIME=	814	13:34	TEMP=	-36.9	0.4079	9726.4
TIME=	851	14:11	TEMP=	-41.8	0.3507	9754.0
TIME=	930	15:30	TEMP=	-45.2	0.3123	9826.3
TIME=	1129	18:49	TEMP=	-46.0	0.3044	10044.9
TIME=	1211	20:11	TEMP=	-50.9	0.2576	10153.8
TIME=	1392	23:12	TEMP=	-49.5	0.2693	10401.1
TIME=	257	4:17	REF=	9409		
TIME=	352	5:52	REF=	9439		
TIME=	400	6:40	REF=	9461		
TIME=	448	7:28	REF=	9484		
TIME=	496	8:16	REF=	9453		
TIME=	545	9: 5	REF=	9540		
TIME=	593	9:53	REF=	9564		
TIME=	642	10:42	REF=	9599		
TIME=	691	11:31	REF=	9632		
TIME=	740	12:20	REF=	9668		
TIME=	789	13: 9	REF=	9707		
TIME=	838	13:58	REF=	9745		
TIME=	887	14:47	REF=	9779		
TIME=	937	15:37	REF=	9834		
TIME=	987	16:27	REF=	9887		
TIME=	1036	17:16	REF=	9935		
TIME=	1086	18: 6	REF=	9995		
TIME=	1136	18:56	REF=	10053		
TIME=	1187	19:47	REF=	10121		
TIME=	1288	21:28	REF=	10259		

LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AUASS
GEOMETRIC ALTITUDES
BASE LEVEL PRESSURE 16.80 MB, ROCKET TEMP -49.97 RAOB TEMP -50.00 (DEG-C)
GEOMETRIC ALT 28000. METERS, GEOPOTENTIAL ALT 27943. METERS

1 KM LEVEL THERMO DATA PRINTOUT

TIME	ALTITUDE(MSL)		WIND VELOCITY(M/SEC)			DIR	FL VEL	WND SHEAR	CORR TEMPS			PRES	DENS	SP SND
(SEC)	(KM)	(KFT)	-N+S	-E+W	TOTAL	(DEG)	(M/SC)	(MPS/M)	(CENT)	(KEL)	CORR	(MB)	(G/CU M)	(M/S)
167.0	63.0	206.7	-16.4	28.8	33.2	299.7	-133.	0.000	-40.9	232.3	-8.4	0.131	0.196	305.6
174.5	62.0	203.4	-6.3	28.7	29.4	282.4	-123.	0.010	-39.3	233.9	-7.1	0.151	0.225	306.7
183.0	61.0	200.1	-6.2	23.3	24.1	284.8	-113.	0.005	-37.8	235.4	-6.1	0.174	0.258	307.7
192.0	60.0	196.9	-4.4	21.9	22.3	281.3	-106.	0.002	-36.6	236.6	-5.3	0.201	0.295	308.5
202.0	59.0	193.6	-2.3	22.6	22.7	275.8	-99.	0.002	-32.7	240.5	-3.8	0.231	0.335	311.0
212.5	58.0	190.3	-2.8	25.0	25.1	276.5	-91.	0.002	-29.1	244.1	-3.1	0.265	0.379	313.3
224.0	57.0	187.0	-0.6	19.1	19.1	271.9	-85.	0.006	-25.3	247.9	-2.6	0.304	0.428	315.7
236.0	56.0	183.7	0.7	12.3	12.3	266.6	-81.	0.007	-21.6	251.6	-2.3	0.348	0.482	318.1
249.0	55.0	180.4	-3.3	3.0	4.4	317.1	-76	0.010	-17.5	255.7	-1.9	0.397	0.541	320.7
263.0	54.0	177.2	-6.9	6.4	9.4	317.0	-71.	0.005	-13.2	260.0	-1.6	0.453	0.607	323.3
277.0	53.0	173.9	-16.2	12.5	20.5	322.4	-67.	0.011	-12.9	260.2	-1.9	0.515	0.690	323.5
293.0	52.0	170.6	-22.5	19.5	29.8	319.1	-63.	0.009	-12.3	260.8	-1.7	0.587	0.783	323.9
309.0	51.0	167.3	-19.5	19.3	27.5	315.2	-59.	0.003	-11.8	261.4	-1.5	0.667	0.889	324.2
326.0	50.0	164.0	-21.9	22.1	31.1	314.7	-56.	0.004	-11.2	261.9	-1.4	0.759	1.010	324.6
344.0	49.0	160.8	-24.2	29.2	37.9	309.6	-53.	0.007	-12.7	260.5	-1.4	0.864	1.155	323.7
364.0	48.0	157.5	-20.3	24.9	32.1	309.2	-50.	0.006	-14.5	258.7	-1.3	0.983	1.324	322.5
384.0	47.0	154.2	-17.7	19.8	26.5	311.7	-46.	0.006	-18.7	254.4	-1.3	1.122	1.536	319.9
407.0	46.0	150.9	-14.8	14.1	20.4	316.4	-43.	0.006	-21.0	252.2	-1.0	1.281	1.770	318.5
431.0	45.0	147.6	-11.2	14.4	18.3	307.8	-39.	0.004	-22.8	250.4	-0.9	1.466	2.039	317.3
458.0	44.0	144.4	-13.8	14.5	20.0	313.5	-36.	0.003	-21.9	251.3	-0.6	1.677	2.325	317.9
487.0	43.0	141.1	-12.6	18.1	22.0	304.8	-33.	0.004	-22.2	251.0	-0.7	1.918	2.663	317.7
519.0	42.0	137.8	-7.6	22.0	23.2	289.1	-30.	0.006	-27.7	245.4	-0.8	2.198	3.120	314.2
553.0	41.0	134.5	-5.5	19.5	20.2	285.7	-28.	0.003	-33.1	240.1	-0.6	2.526	3.666	310.7
590.0	40.0	131.2	-3.7	14.0	14.5	284.7	-26.	0.006	-32.4	240.7	-0.4	2.908	4.208	311.2
640.0	39.0	128.0	-4.7	12.8	13.6	290.2	-24.	0.002	-30.1	243.0	-0.3	3.344	4.794	312.6
673.0	38.0	124.7	-2.9	14.1	14.4	281.5	-22.	0.002	-33.6	239.5	-0.4	3.848	5.596	310.4
719.0	37.0	121.4	-1.1	11.8	11.8	275.5	-21.	0.003	-35.4	237.8	-0.3	4.434	6.497	309.2
766.0	36.0	118.1	-1.2	9.3	9.4	277.3	-20.	0.002	-36.3	236.9	-0.3	5.114	7.521	308.7
818.0	35.0	114.8	-3.9	10.3	11.0	290.8	-18.	0.003	-37.8	235.4	-0.4	5.902	8.736	307.7
880.0	34.0	111.5	-1.9	8.7	8.9	282.3	-16.	0.003	-43.3	229.8	-0.3	6.827	10.349	304.0
945.0	33.0	108.3	-1.4	6.7	6.9	281.8	-15.	0.002	-45.5	227.7	-0.2	7.917	12.116	302.6
1010.0	32.0	105.0	-3.2	9.2	9.7	289.2	-14.	0.003	-45.7	227.4	-0.2	9.189	14.076	302.4
1087.0	31.0	101.7	-3.9	10.2	10.9	291.1	-13.	0.001	-46.0	227.1	-0.2	10.667	16.362	302.2
1167.0	30.0	98.4	-1.9	8.6	8.8	282.2	-12.	0.003	-48.5	224.7	-0.2	12.395	19.221	300.6
1203.0	29.0	95.1	0.6	5.9	5.9	263.7	-11.	0.004	-50.7	222.4	-0.2	14.427	22.597	299.1
1352.0	28.0	91.9	-0.4	7.4	7.4	272.9	-10.	0.002	-50.0	223.2	-0.2	16.800	26.223	299.6

LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AUASS
GEOMETRIC ALTITUDES
BASE LEVEL PRESSURE 16.80 MB. ROCKET TEMP -49.97 RAOB TEMP -50.00 (DEG-C)
GEOMETRIC ALT 28000. METERS. GEOPOTENTIAL ALT 27943. METERS

SIGNIFICANT LEVEL THERMO DATA

TIME (SEC)	ALTITUDE(MSL) (KM)	(KFT)	WIND VELOCITY(M/SEC) -N+S	-E+W	TOTAL	DIR (DEG)	FL VEL (M/SC)	WIND SHEAR (MPS/M)	CORR TEMPS (CENT)	(KEL)	CORR	PRES (MB)	DENS (G/CU M)	SP SND (M/S)
194.0	59.8	196.2	-4.0	22.1	22.4	280.2	-104.	0.000	-35.8	237.4	-5.0	0.207	0.303	309.0
264.0	53.9	176.9	-7.5	6.9	10.2	317.4	-71.	0.016	-13.2	260.0	-1.6	0.457	0.613	323.4
329.0	49.8	163.5	-22.3	23.3	32.2	313.8	-56.	0.022	-11.5	261.7	-1.4	0.776	1.034	324.4
365.0	47.9	157.3	-20.2	24.6	31.9	309.3	-50.	0.003	-14.7	258.5	-1.3	0.990	1.335	322.4
389.0	46.8	153.5	-17.1	18.5	25.2	312.7	-46.	0.007	-19.2	253.9	-1.3	1.156	1.587	319.6
437.0	44.8	146.9	-11.7	14.5	18.6	309.0	-38.	0.007	-22.6	250.6	-0.8	1.513	2.103	317.5
473.0	43.5	142.7	-13.3	16.4	21.1	309.0	-34.	0.002	-22.0	251.1	-0.6	1.802	2.500	317.8
503.0	42.5	139.4	-10.2	20.2	22.6	296.9	-32.	0.005	-25.0	248.2	-0.7	2.058	2.891	315.9
533.0	41.6	136.4	-6.7	21.0	22.0	287.7	-30.	0.004	-29.9	243.2	-0.7	2.333	3.345	312.8
564.0	40.7	133.5	-4.9	17.9	18.5	285.4	-28.	0.004	-32.9	240.3	-0.5	2.640	3.827	310.9
636.0	38.9	127.5	-4.5	13.0	13.7	289.0	-24.	0.005	-30.6	242.5	-0.3	3.415	4.906	312.3
685.0	37.7	123.8	-2.4	13.5	13.7	279.9	-22.	0.002	-34.1	239.1	-0.4	4.001	5.831	310.1
814.0	35.1	115.1	-3.7	10.2	10.9	289.8	-18.	0.004	-37.7	235.5	-0.4	5.841	8.643	307.7
851.0	34.5	113.1	-2.8	9.5	9.9	286.3	-17.	0.001	-40.7	232.4	-0.3	6.395	9.595	305.7
930.0	33.2	109.0	-1.5	7.2	7.3	281.9	-16.	0.003	-45.0	228.2	-0.2	7.666	11.708	302.9
1129.0	30.5	100.0	-2.8	9.4	9.8	286.5	-12.	0.003	-47.3	225.8	-0.2	11.574	17.863	301.4
1211.0	29.5	96.7	-0.4	7.3	7.3	272.8	-11.	0.003	-49.7	223.5	-0.2	13.434	20.949	299.8

```
 LOKI          NUMBER   73 LAUNCHED 21 SEP 76  705 FROM POKER FLAT
STATION LATITUDE 65 117 LONGITUDE    147 459 ALTITUDE  412  METERS SYSTEM AUASS
 GEOMETRIC ALTITUDES
BASE LEVEL PRESSURE 16 80 MB, ROCKET TEMP -49 97 RAOB TEMP -50 00 (DEG-C)
GEOMETRIC ALT 28000  METERS,     GEOPOTENTIAL ALT 27943  METERS

     ***MRN - 30 CARDS***

  STA  DATE TIME ALT DIR SP UNS UFN NS  EW FV TMP COR  PRES   DEN  S/S
 70192760921070506300300033-013 019-016 029133-041-081. 305-11. 958-1306      30
 70192760921070506200287029-010 021-006 029123-039-071. 508-12. 246-1307      30
 70192760921070506100285024-007 023-006 023113-038-061. 740-12. 575-1308      30
 70192760921070506000281022-006 022-004 022106-037-052. 006-12. 953-1308      30
 70192760921070505980280022-006 021-004 022104-036-052. 066-13. 032-1309      30
 70192760921070505900276023-007 020-002 023099-033-042. 309-13. 346-1311      30
 70192760921070505800276025-008 021-003 025091-029-032. 653-13. 787-1313      30
 70192760921070505700272019-005 019-001 019085-025-033. 042-14. 275-1316      30
 70192760921070505600267012-002 014 001 012081-022-023. 480-14. 819-1318      30
 70192760921070505500317004-002 006-003 003076-017-023. 974-15. 415-1321      30
 70192760921070505400317009-006 005-007 006071-013-024. 528-16. 069-1323      30
 70192760921070505397000010-006 006-007 007071-013-024. 573-16. 128-1323      30
 70192760921070505300000020-014 010-016 012067-013-025. 154-16. 900-1324      30
 70192760921070505200000030-021 017-023 019063-012-025. 866-17. 834-1324      30
 70192760921070505100315027-021 019-019 019059-012-016. 673-18. 894-1324      30
 70192760921070505000315031-022 021-022 022056-011-017. 591-11. 010+0325      30
 70192760921070504983314032-023 022-022 023056-011-017. 765-11. 034+0324      30
 70192760921070504900310038-024 026-024 029053-013-018. 636-11. 155+0324      30
 70192760921070504800309032-021 025-020 025050-014-019. 835-11. 324+0323      30
 70192760921070504795309032-021 025-020 025050-015-019. 904-11. 335+0322      30
 70192760921070504700312027-019 021-018 020046-019-011. 122+01. 536+0320      30
 70192760921070504678313025-018 019-017 019046-019-011. 156+01. 587+0320      30
 70192760921070504600000020-015 015-015 014043-021-011. 281+01. 770+0318      30
 70192760921070504500308018-011 014-011 014039-023-011. 466+02. 039+0317      30
 70192760921070504478309019-012 014-012 014038-023-011. 513+02. 103+0317      30
 70192760921070504400000020-013 014-014 015036-022-011. 677+02. 325+0318      30
 70192760921070504348309021-013 016-013 016034-022-011. 802+02. 500+0318      30
 70192760921070504300305022-013 018-013 018033-022-011. 918+02. 663+0318      30
 70192760921070504250297023-011 020-010 020032-025-012. 058+02. 891+0316      30
 70192760921070504200289023-008 022-008 022030-028-012. 198+03. 120+0314      30
 70192760921070504159288022-007 021-007 021030-030-012. 333+03. 345+0313      30
 70192760921070504100000020-006 020-005 019028-033-012. 526+03. 666+0311      30
 70192760921070504070285019-005 018-005 018028-033-012. 640+03. 827+0311      30
 70192760921070504000285014-004 014-004 014026-032 002. 908+04. 208+0311      30
 70192760921070503900290014-005 013-005 013024-030 003. 344+04. 794+0313      30
 70192760921070503886289014-004 013-004 013024-031 003. 415+04. 906+0312      30
 70192760921070503800282014-003 014-003 014022-034 003. 848+05. 596+0310      30
 70192760921070503774280014-003 013-002 014022-034 004. 001+05. 831+0310      30
 70192760921070503700275012-001 012-001 012021-035 004. 434+06. 497+0309      30
 70192760921070503600277009-001 009-001 009020-036 005. 114+07. 521+0309      30
 70192760921070503508290011-004 010-004 010018-038 005. 841+08. 643+0308      30
 70192760921070503500291011-004 010-004 010018-038 005. 902+08. 736+0308      30
 70192760921070503447000010-003 009-003 010017-041 006. 395+09. 595+0306      30
 70192760921070503400282009-002 009-002 009016-043 006. 827+01. 035+1304      30
 70192760921070503323282007-001 007-002 007016-045 007. 666+01. 171+1303      30
 70192760921070503300282007-001 007-001 007015-046 007. 917+01. 212+1303      30
 70192760921070503200000010-003 009-003 009014-046 009. 189+01. 408+1302      30
 70192760921070503100291011-004 010-004 010013-046 001. 067+11. 636+1302      30
 70192760921070503047000010-003 009-003 009012-047 001. 157+11. 786+1301      30
 70192760921070503000282009-002 009-002 009012-049 001. 240+11. 922+1301      30
 70192760921070502949273007-001 007 000 007011-050 001. 343+12. 095+1300      30
 70192760921070502900264006 001 006 001 006011-051 001. 443+12. 260+1299      30
 70192760921070502800273007 000 007 000 007010-050 001. 680+12. 622+1300      30
```

LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AUASS
GEOPOTENTIAL ALTITUDES
BASE LEVEL PRESSURE 16.80 MB. ROCKET TEMP -49.97 RAOB TEMP -50.00 (DEG-C)
GEOMETRIC ALT 28000 METERS, GEOPOTENTIAL ALT 27943. METERS

MANDATORY LEVEL THERMO DATA

TIME	ALTITUDE(MSL)		WIND VELOCITY(M/SEC)			DIR	FL VEL	WIND SHEAR		CORR TEMPS		PRES	DENS	SP SND
(SEC)	(KM)	(KFT)	-N+S	-E+W	TOTAL	(DEG)	(M/SC)	(MPS/M)	(CENT)	(KEL)	CORR	(MB)	(G/CU M)	(M/S)
191.8	60.2	197.4	-4.4	21.9	22.4	281.4	-106.	0.000	-36.6	236.6	-5.3	0.200	0.295	308.5
222.8	57.2	187.8	-0.8	19.7	19.7	272.4	-86.	0.004	-25.7	247.5	-2.7	0.300	0.422	315.5
249.7	55.1	180.7	-3.4	3.2	4.7	317.1	-75.	0.017	-17.3	255.9	-1.9	0.400	0.545	320.8
273.6	53.4	175.1	-13.8	11.1	17.7	321.1	-68.	0.013	-13.0	260.2	-1.9	0.500	0.670	323.5
315.1	50.8	166.5	-20.3	20.3	28.8	315.0	-58.	0.011	-11.6	261.6	-1.4	0.700	0.932	324.4
366.4	48.0	157.5	-20.0	24.3	31.5	309.5	-50.	0.004	-15.0	258.2	-1.3	1.000	1.350	322.2
496.3	42.8	140.4	-11.3	19.3	22.4	300.2	-32.	0.010	-23.8	249.4	-0.7	2.000	2.796	316.7
598.4	39.9	130.8	-3.9	13.8	14.3	285.8	-26.	0.009	-31.9	241.2	-0.4	3.000	4.332	311.5
758.1	36.3	118.9	-1.2	9.7	9.8	277.0	-20.	0.005	-36.1	237.0	-0.3	5.000	7.350	308.7
890.3	33.9	111.3	-1.8	8.4	8.6	282.2	-16.	0.001	-43.7	229.5	-0.3	7.000	10.629	303.8
1052.2	31.5	103.4	-3.6	9.7	10.4	290.3	-13.	0.002	-45.9	227.3	-0.2	10.000	15.330	302.3

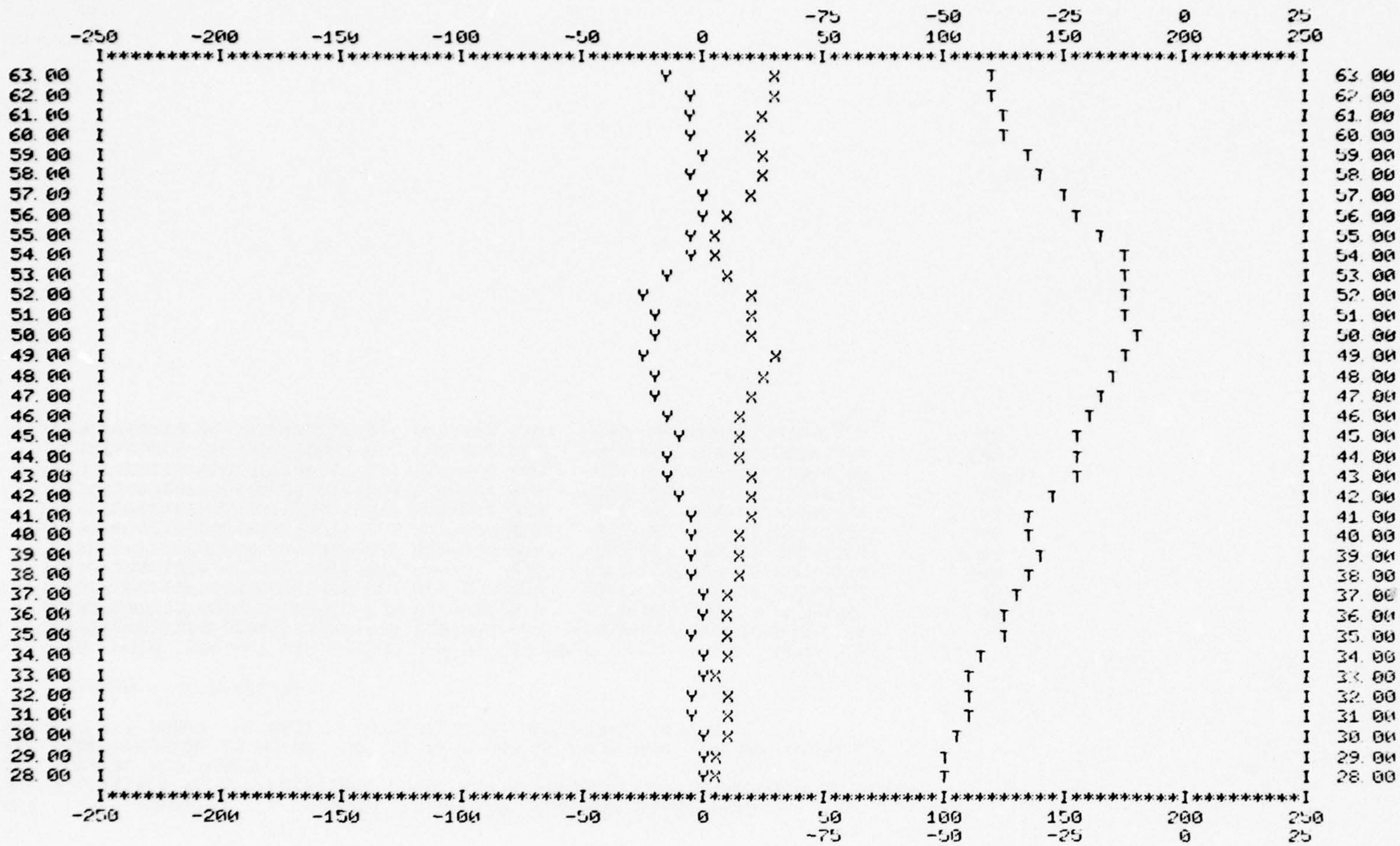
```
LOKI          NUMBER   73 LAUNCHED 21 SEP 76  705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE   147.459 ALTITUDE  412.  METERS SYSTEM AUASS
 GEOPOTENTIAL ALTITUDES
BASE LEVEL PRESSURE 16.80 MB, ROCKET TEMP -49.97 RAOB TEMP -50.00 (DEG-C)
GEOMETRIC ALT 28000.  METERS,     GEOPOTENTIAL ALT 27943.  METERS

      ***MRN - 40 CARDS***

     STA  DATE TIME ALT DIR SP UNS UEW NS  EW FV TMP COR  PRES    DEN  S/S
    70192760921070506016281022-006 022-004 022   -037-052.000-12.945-1308          40
    70192760921070505724000020-005 019-001 020   -026-033.000-14.223-1315          40
    70192760921070505508317005-002 006-003 003   -017-024.000-15.446-1321          40
    70192760921070505337321018-012 009-014 011   -013-025.000-16.695-1323          40
    70192760921070505076315029-021 020-020 020   -012-017.000-19.322-1324          40
    70192760921070504799310031-021 025-020 024   -015-011.000+01.350+0322          40
    70192760921070504281300022-011 019-011 019   -024-012.000+02.796+0317          40
    70192760921070503988286014-004 014-004 014   -032 003.000+04.332+0311          40
    70192760921070503625000010-001 010-001 010   -036 005.000+07.350+0309          40
    70192760921070503392282009-002 008-002 008   -044 007.000+01.063+1304          40
    70192760921070503152000010-004 010-004 010   -046 001.000+11.533+1302          40
```

```
LOKI          NUMBER   73 LAUNCHED 21 SEP 76  705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE    147.459 ALTITUDE  412. METERS SYSTEM AVASS
 GEOMETRIC ALTITUDES
BASE LEVEL PRESSURE 16.80 MB. ROCKET TEMP -49.97 RAOB TEMP -50.00 (DEG-C)
GEOMETRIC ALT 28000. METERS,     GEOPOTENTIAL ALT 27943. METERS
```

*** X AND Y WIND(M/S) AND TEMP(DEG-C) . VS. ALT(KM) ***



LOKI NUMBER 73 LAUNCHED 21 SEP 76 705 FROM POKER FLAT
STATION LATITUDE 65.117 LONGITUDE 147.459 ALTITUDE 412. METERS SYSTEM AUHSS
GEOMETRIC ALTITUDES

ROCOB MESSAGE

KM	WIND SPEED	WIND DIR	TEMP	WS	WD	TEMP		ROCOB	
28	7.4	272.9	-50.0				28550	27007	91262
30	8.8	282.2	-48.5				30549	28009	91192
33	6.9	281.8	-45.5			X	33546	28007	91121
35	11.0	290.8	-37.8				35538	29011	92874
39	13.6	290.2	-30.1			X	39530	29014	92479
40	14.5	284.7	-32.4				40532	28014	92421
41	20.2	285.7	-33.1	X			41533	29020	92367
43	22.0	304.8	-22.2			X	43522	30022	92266
45	18.3	307.8	-22.8				45523	31018	92204
48	32.1	309.2	-14.5	X			48514	31032	92132
49	37.9	309.6	-12.7	X			49513	31038	92116
50	31.1	314.7	-11.2				50511	31031	92101
52	29.8	319.1	-12.3	X			52512	32030	93783
55	4.4	317.1	-17.5				55517	32004	93541
57	19.1	271.9	-25.3	X			57525	27019	93428
60	22.3	281.3	-36.6				60537	28022	93295
63	33.2	299.7	-40.9				63541	30033	93196

11046	10315	29010
11044	07339	28009
11036	05363	28010
11032	03399	29014
11024	02428	30022
11015	01480	31031
22012	07508	32029
22013	05534	32018
22017	04551	32005
22026	03572	27020
22037	02602	28022

D. LOKI CALIBRATION TAPE PREPARATION

On page 35 is a listing of the 1st page of the calibration data for the LOKI flight just processed. The data that must be punched on paper tape is contained in column 2 and the last row as indicated by the dashed lines. The tape is punched by typing on the teletype as follows:(Note the 1st '*' is typed by the computer, the 2nd '*' is typed by the operator)

```
*BU PTRP:
** 0.9911
** 0.9571
  etc
**0.1472
**0.1333
** 1199.000
** 430.000
** 96.160
** 26.920
** 6.110
*$EXIT
*ENDB
```

The tape just prepared is then used when the program requests--

"PUT LOKI TEMP TAPE IN READER PRESS CR"

LOKI CALIBRATION DATA

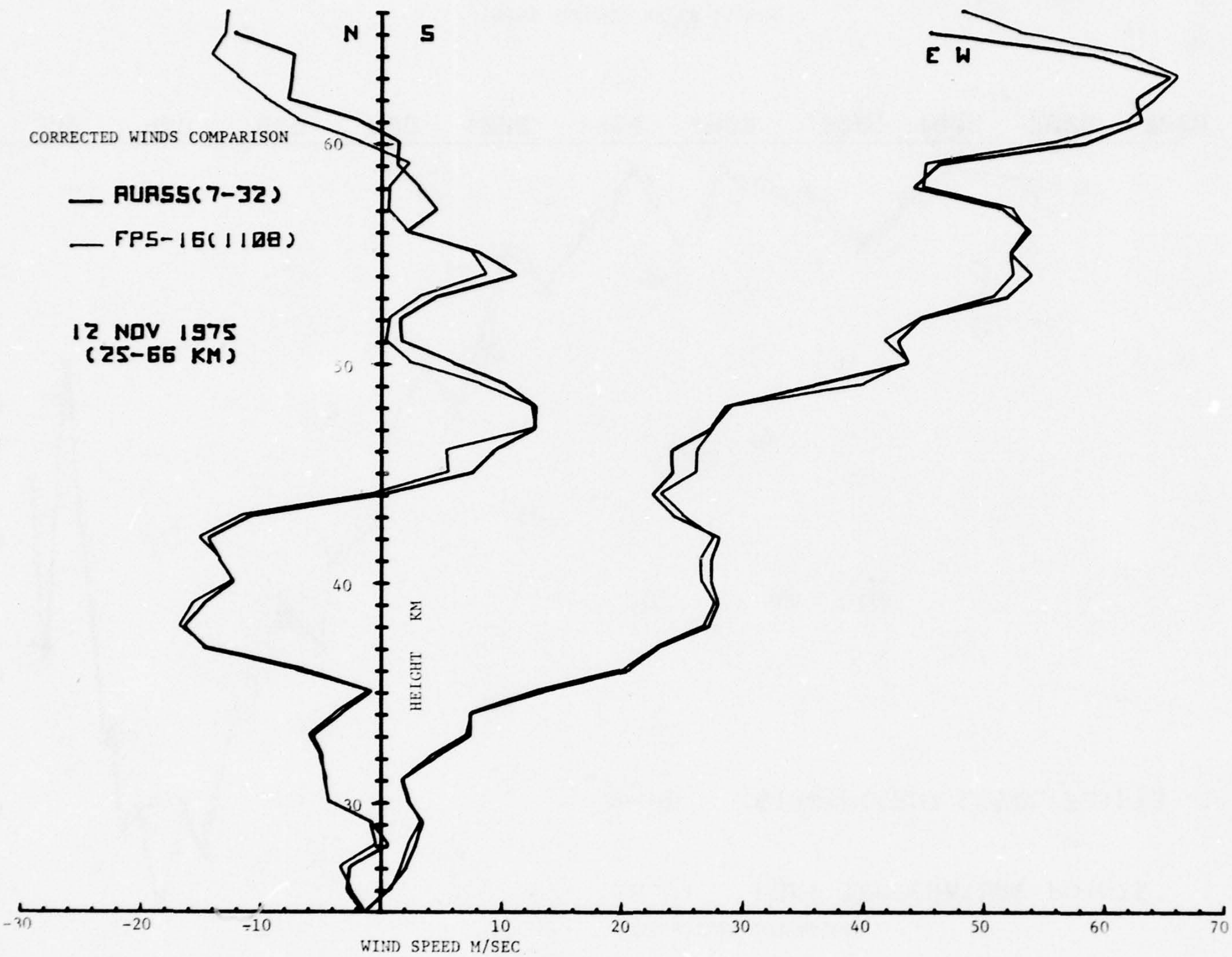
INSTRUMENT NØ20413 THERMISTOR NØ12886

21	5			
1.0	0.9911	5572.0	5622.0	
5.0	0.9571	5572.0	5822.0	
10.0	0.9186	5573.0	6067.0	
20.0	0.8505	5573.0	6553.0	
30.0	0.7930	5573.0	7028.0	
40.0	0.7435	5573.0	7496.0	
50.0	0.7001	5574.0	7962.0	
60.0	0.6618	5574.0	8422.0	
70.0	0.6293	5575.0	8859.0	
80.0	0.5994	5575.0	9301.0	
90.0	0.5730	5575.0	9729.0	
100.0	0.5502	5576.0	10135.0	
150.0	0.4589	5576.0	12150.0	
200.0	0.3981	5577.0	14009.0	
300.0	0.3217	5578.0	17341.0	
400.0	0.2753	5579.0	20201.0	
600.0	0.2223	5579.0	25033.0	
800.0	0.1925	5580.0	28836.0	
1000.0	0.1740	5581.0	31883.0	
1500.0	0.1472	5582.0	37417.0	
2000.0	0.1333	5583.0	41258.0	
-65.0	-50.0	-25.0	0.0	35.0
1199.000	430.000	96.160	26.920	6.110

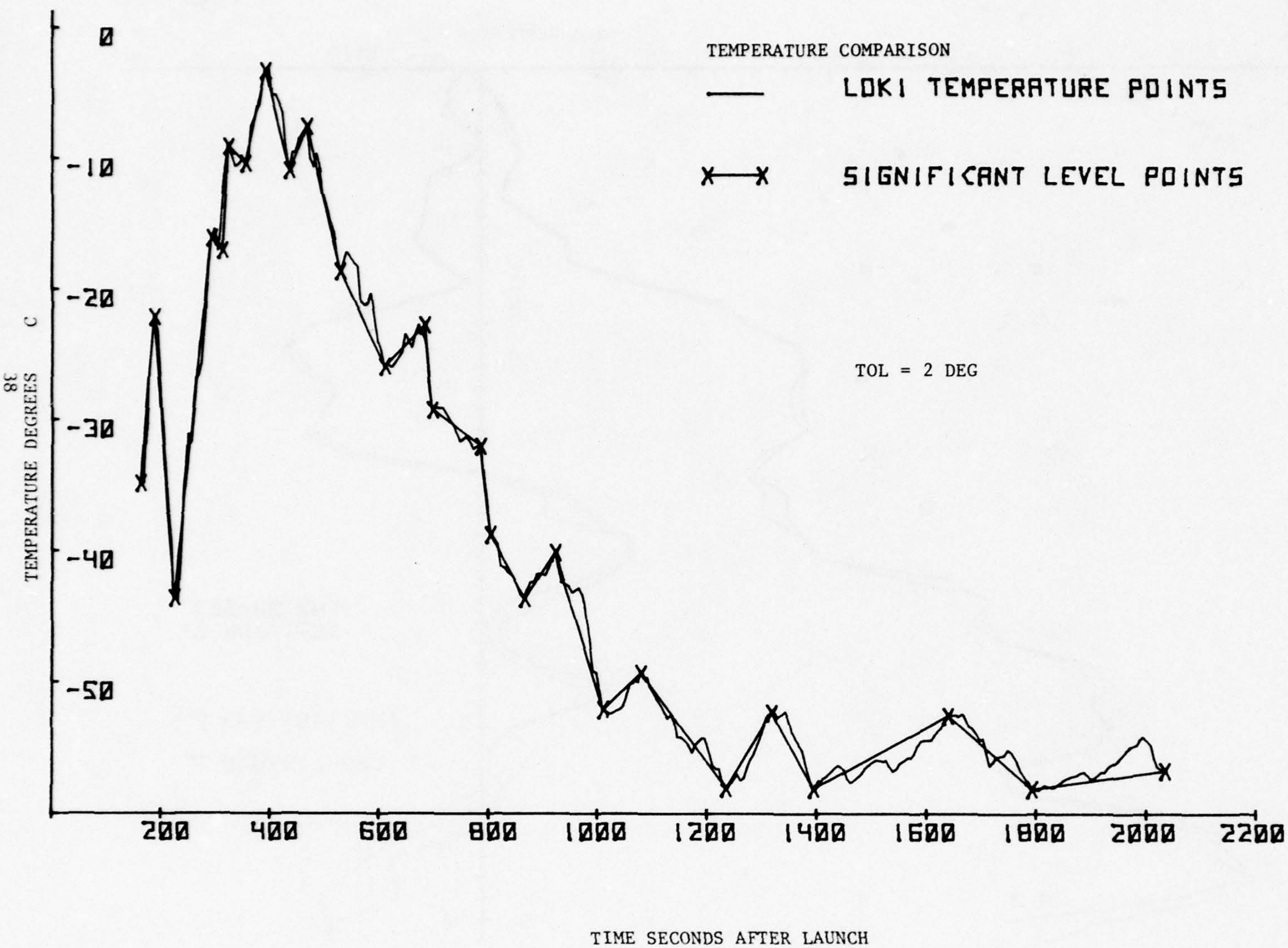
III. ACCURACY COMPARISONS

The accuracy of the Interdata 7-32--Nike Hercules system(labeled AUASS(7-32)) was estimated by comparing the rocketsonde corrected winds from this system to the corrected winds from the UNIVAC 1108--FPS-16 system. The results of this comparison are shown on page 37. Clearly the results are in good agreement, as the shape of the profile for both systems is very similar. Absolute differences are within 2 to 4 M/S at high altitudes and 1/2 to 1 M/S at lower altitudes.

The accuracy of the temperature processing is achieved by comparing the plot of the actual raw temperature data to the significant level temperature data obtained by the computer. This comparison is shown on page 38 for a temperature tolerance of 2 degrees C. The temperature tolerance guarantees that the segmented curve will never be more than 2 degrees from the actual temperature curve. The data selected by the computer does agree very well to the peaks and valleys of the raw temperature profile.



CORRECTED WINDS COMPARISON
AVASS(7-32)
FPS-16(1108)
12 NOV 1975
(25-66 KM)
N
S
E W
60
50
40
30
HEIGHT KM
-30
-20
-10
10
20
30
40
50
60
70
WIND SPEED M/SEC

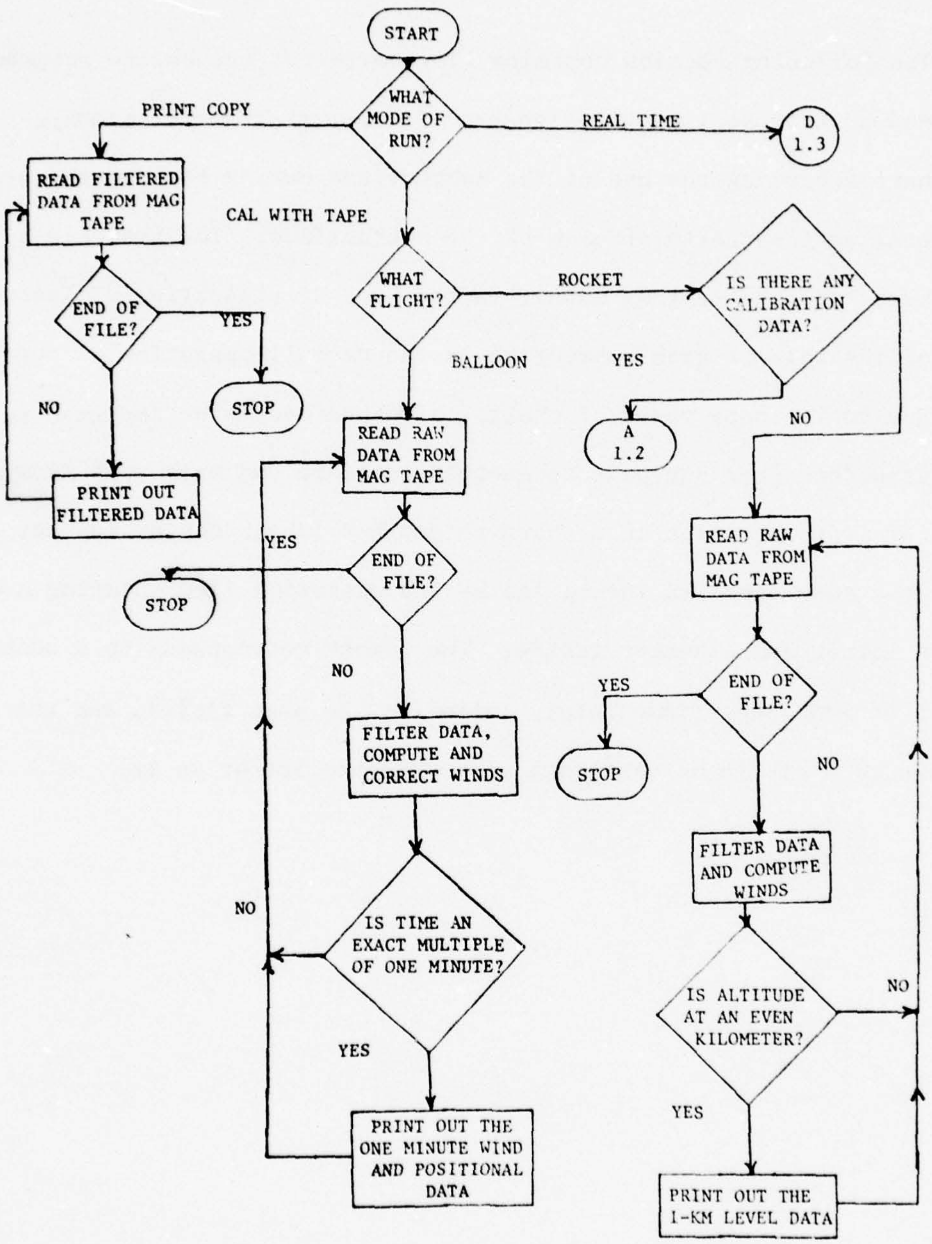

TEMPERATURE COMPARISON
LOKI TEMPERATURE POINTS
SIGNIFICANT LEVEL POINTS
TOL = 2 DEG
0
-10
-20
-30
-40
-50
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
TEMPERATURE DEGREES C
TIME SECONDS AFTER LAUNCH

IV. FLOWCHARTS

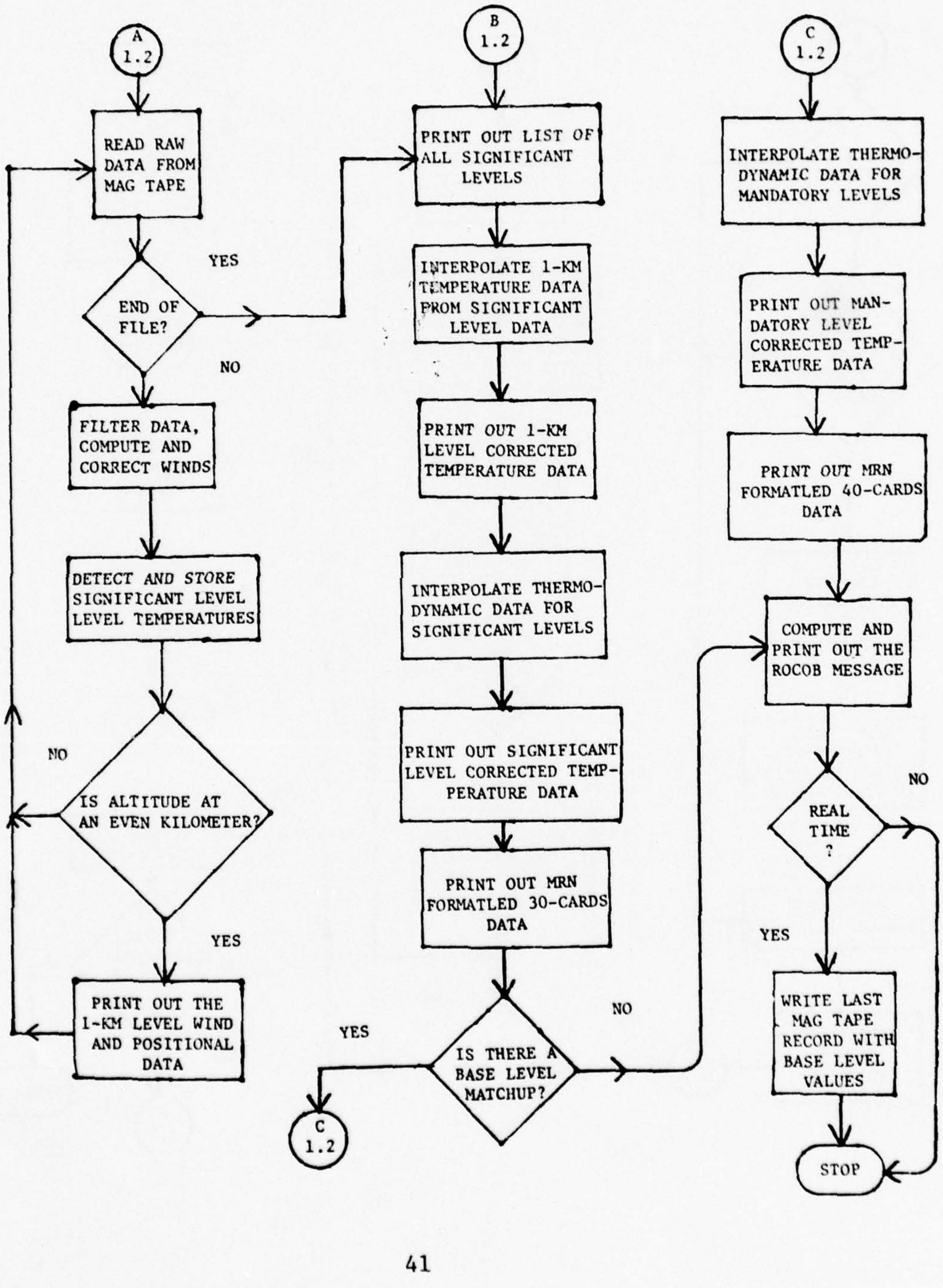
The following section contains flowcharts for the entire program. Included in this section are a general flowchart of the program, a flowchart depicting the use of the subroutines during program execution, and detailed flowcharts of each of the subroutines. The flowcharts have been arranged in this manner in the hope of presenting a clearer view of the role of each subroutine in the overall operation of the program.

Due to the many pages of charts, a method had to be implemented to keep transfers from one page to another separate and easy to follow. A transfer from one point in a chart to another (which may or may not be on the same page) is identified by the execution flow entering a circle with a letter and a number inside. The number corresponds to a number at top of some page (immediately following the page title), and the letter matches to a circle on that page with the same letter in it.

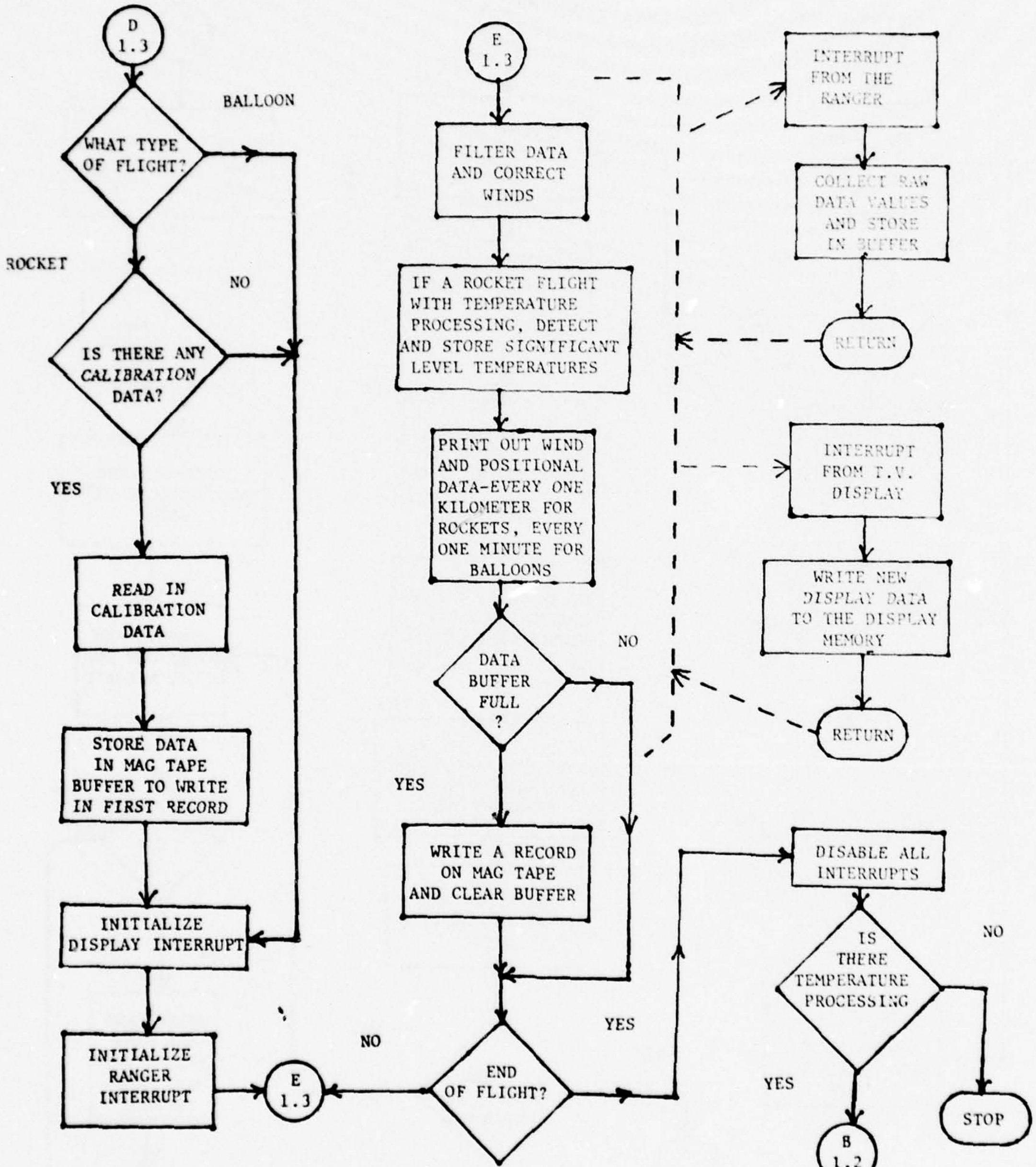
GENERAL FLOWCHART FOR PROGRAM- 1.1



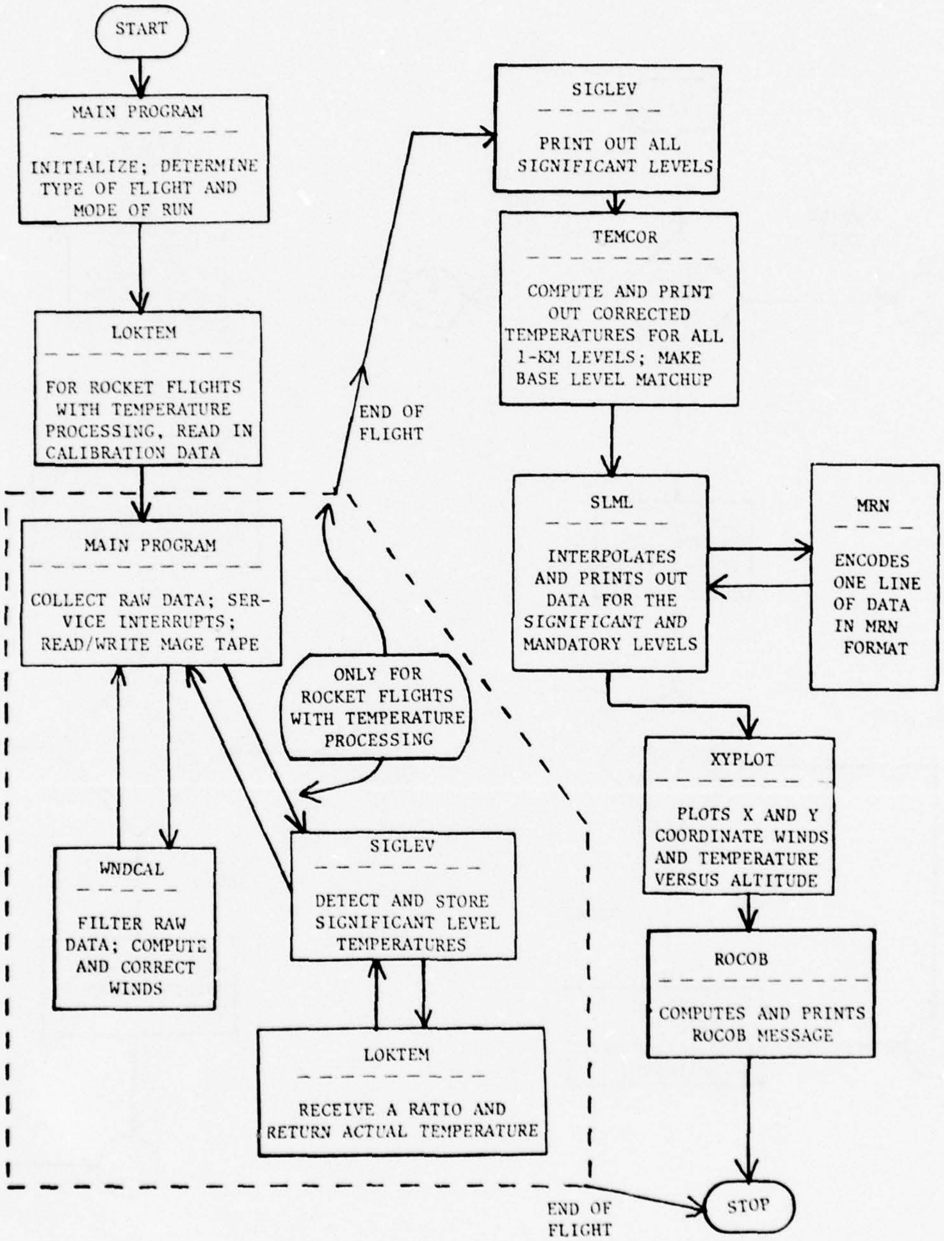
GENERAL FLOWCHART FOR PROGRAM -1.2



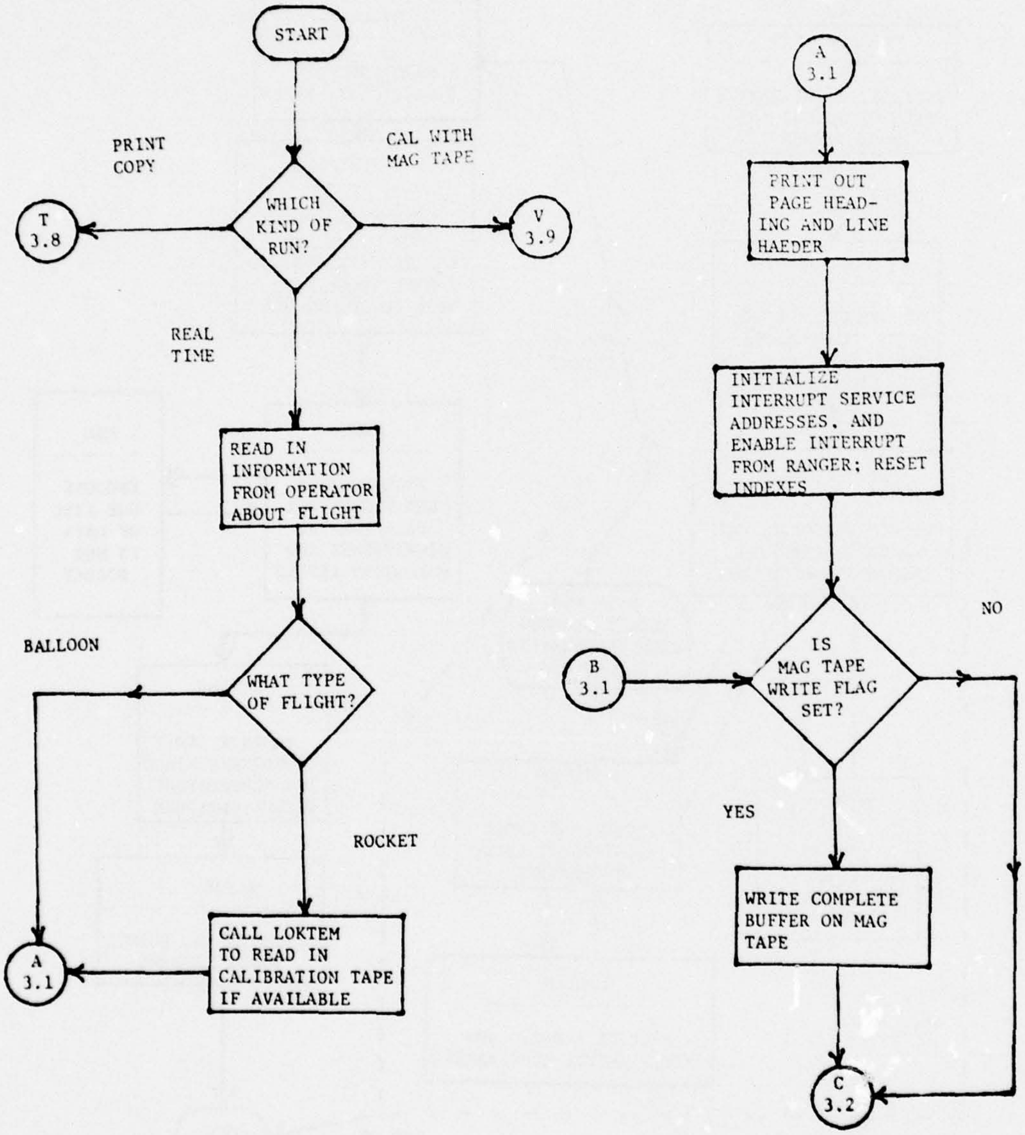
GENERAL FLOWCHART FOR PROGRAM -1.3



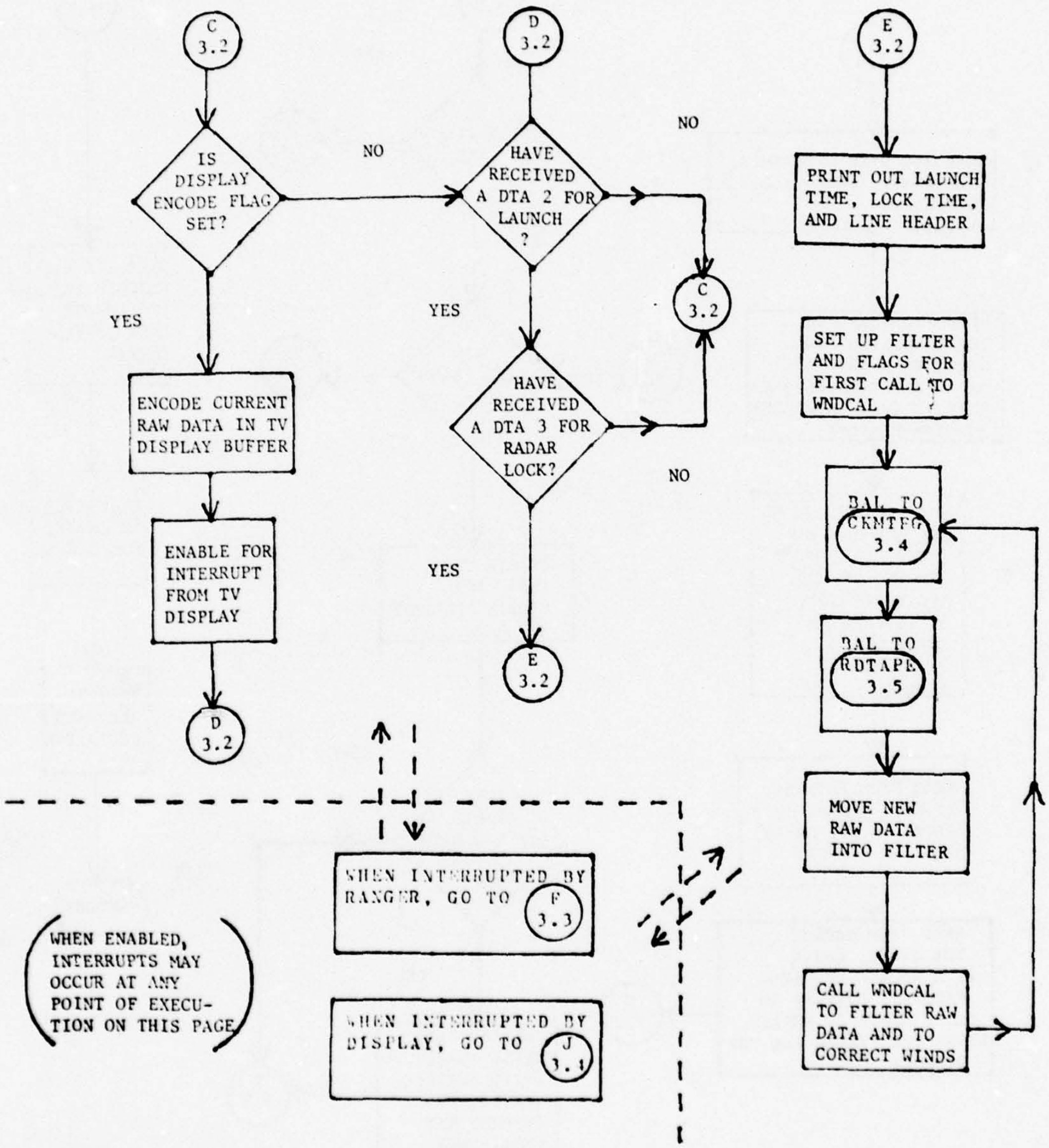
USE OF SUBROUTINES BY PROGRAM -2.1



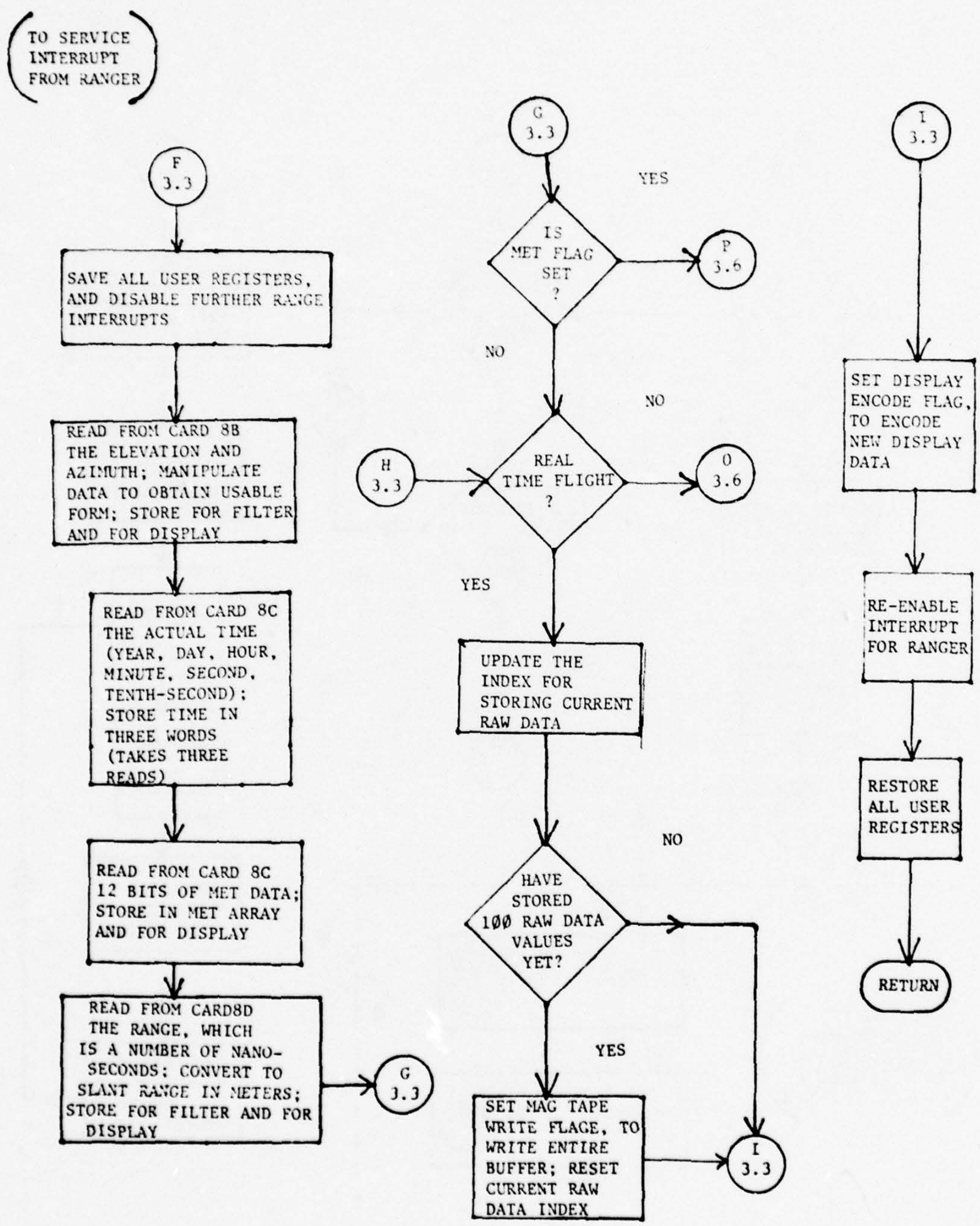
FLOWCHART FOR MAIN PROGRAM - 3.1



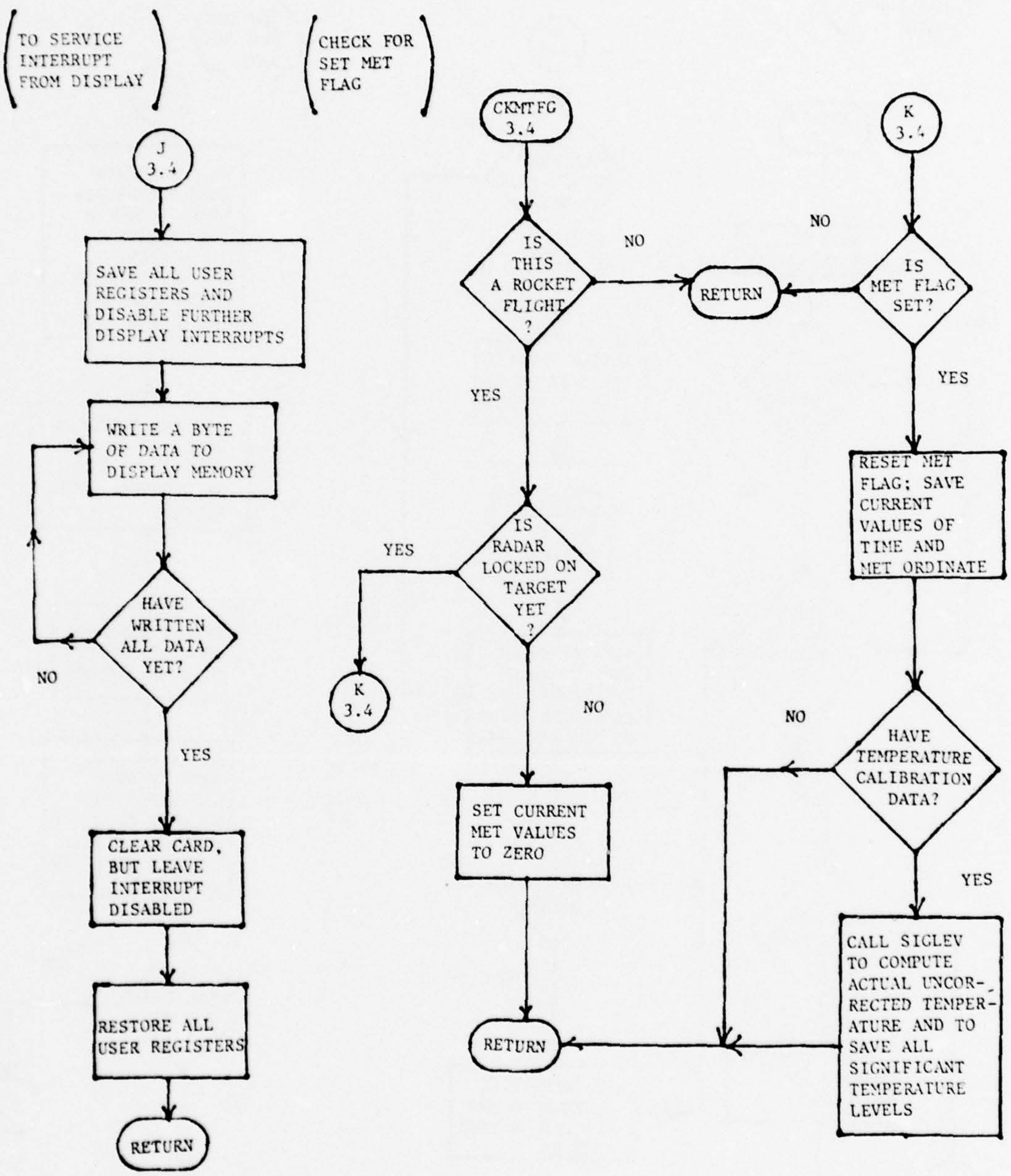
FLOWCHART FOR MAIN PROGRAM - 3.2



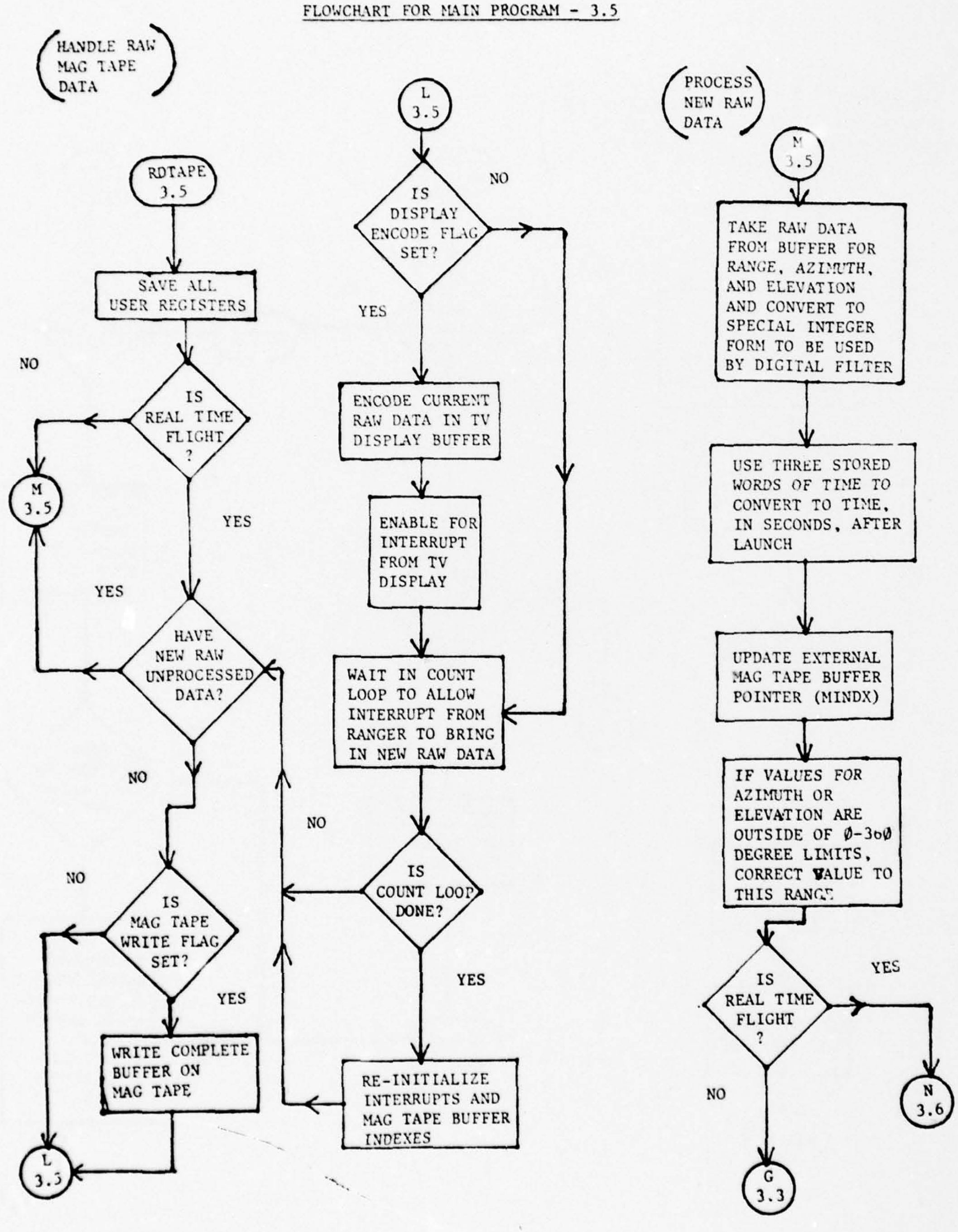
FLOWCHART FOR MAIN PROGRAM - 3.3



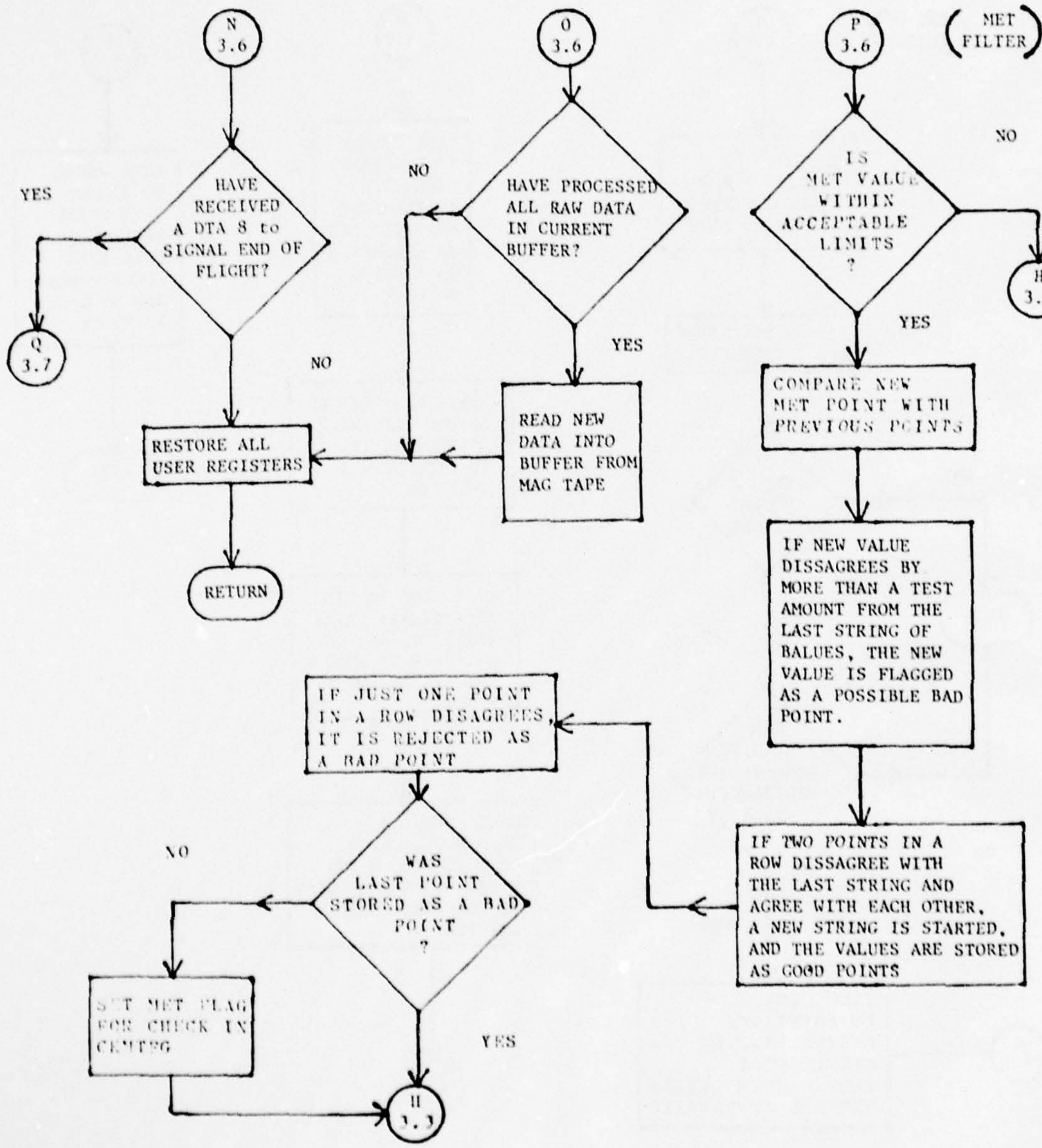
FLOWCHART FOR MAIN PROGRAM - 3.4



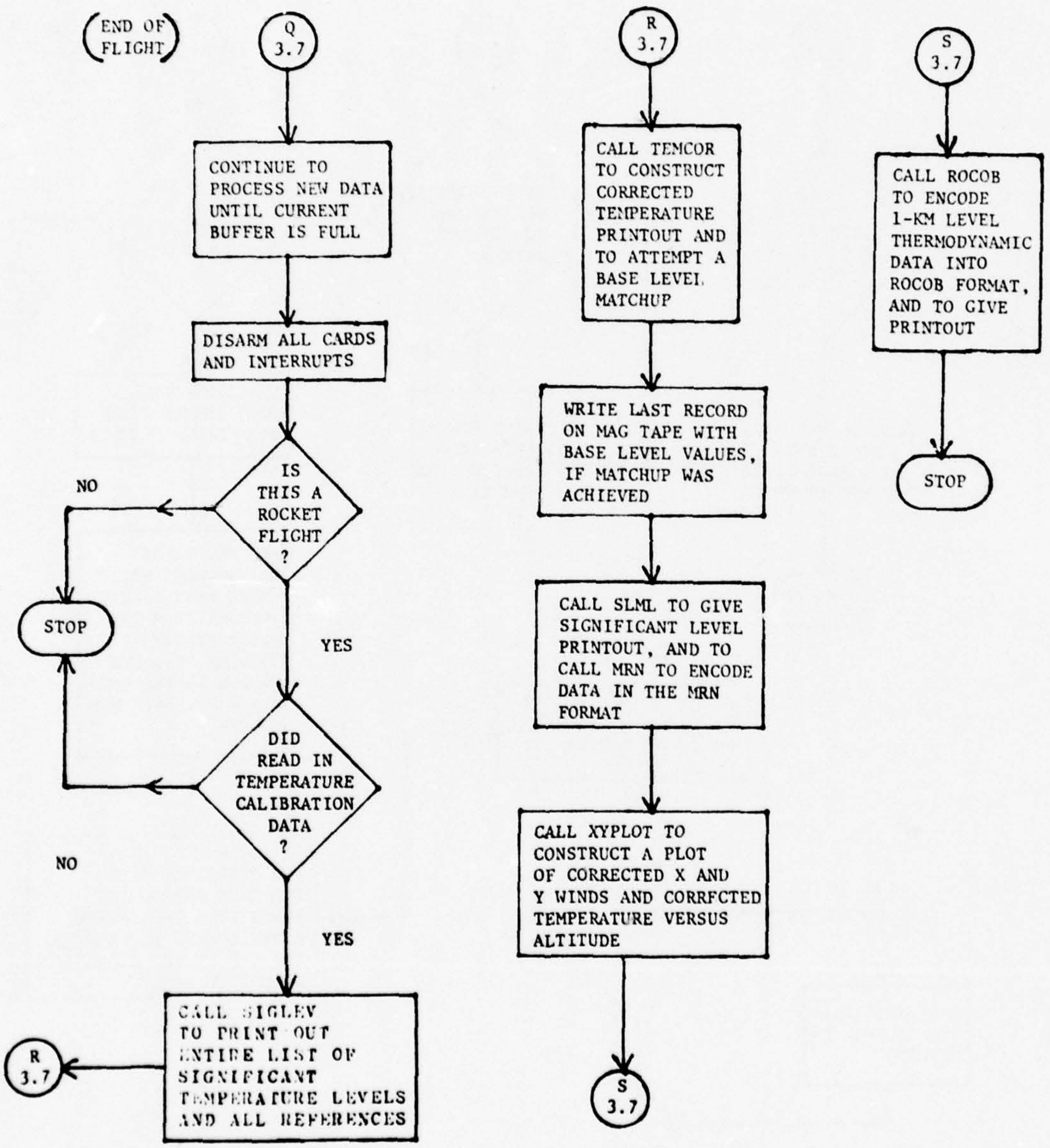
FLOWCHART FOR MAIN PROGRAM - 3.5



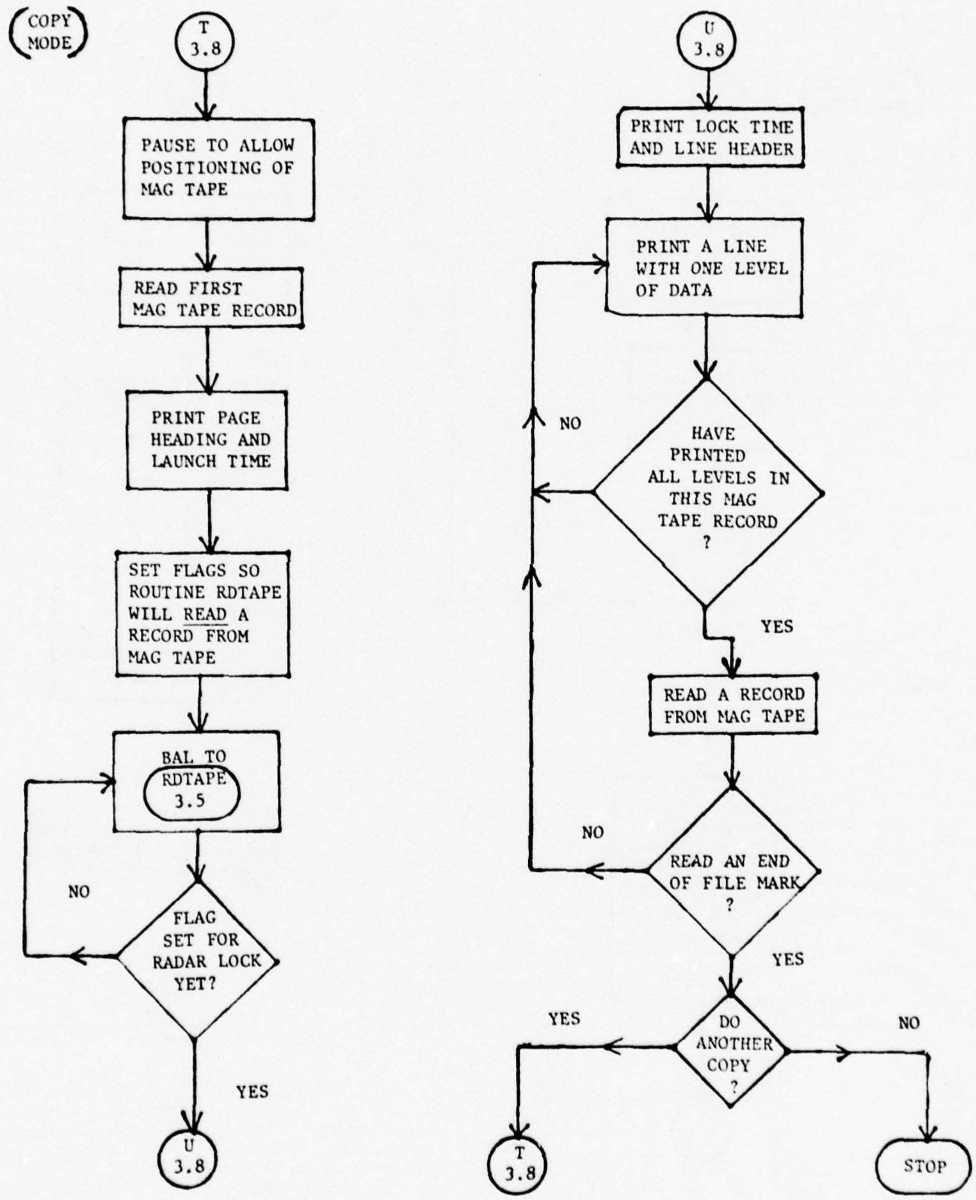
FLOWCHART FOR MAIN PROGRAM - 3.6



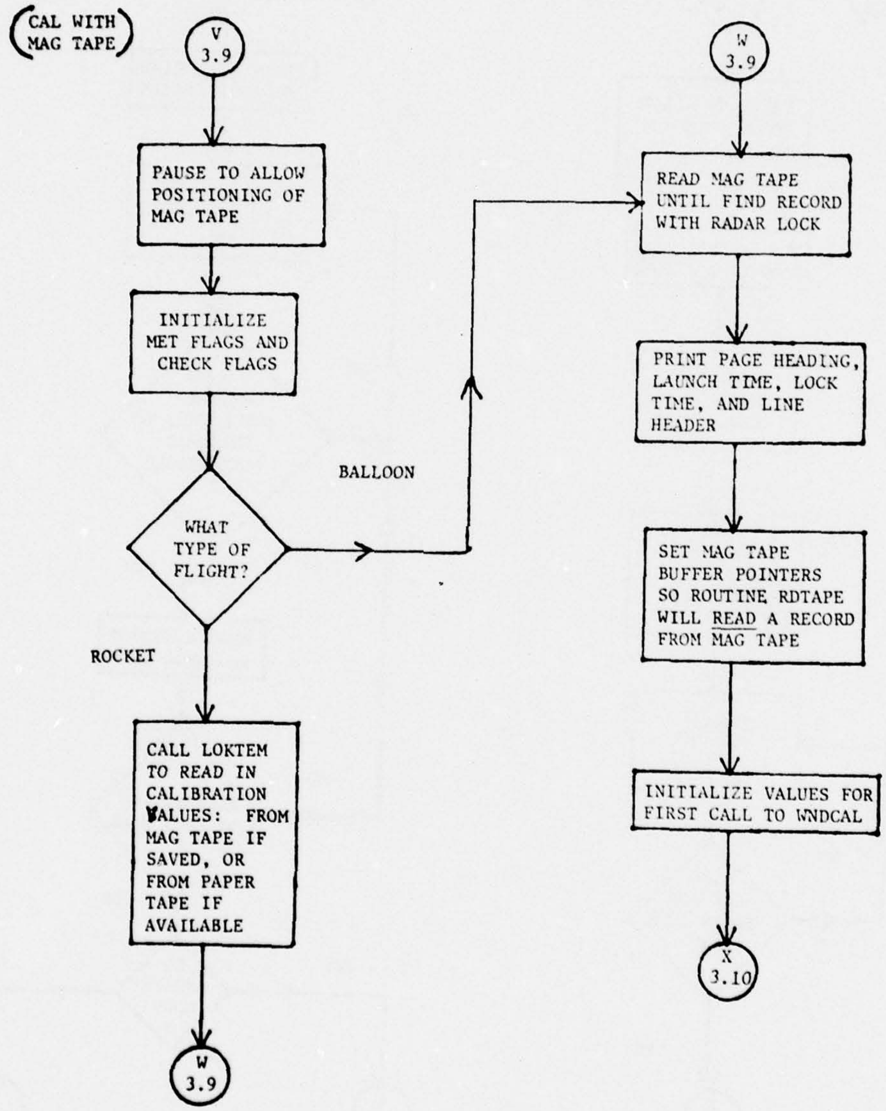
FLOWCHART FOR MAIN PROGRAM - 3.7



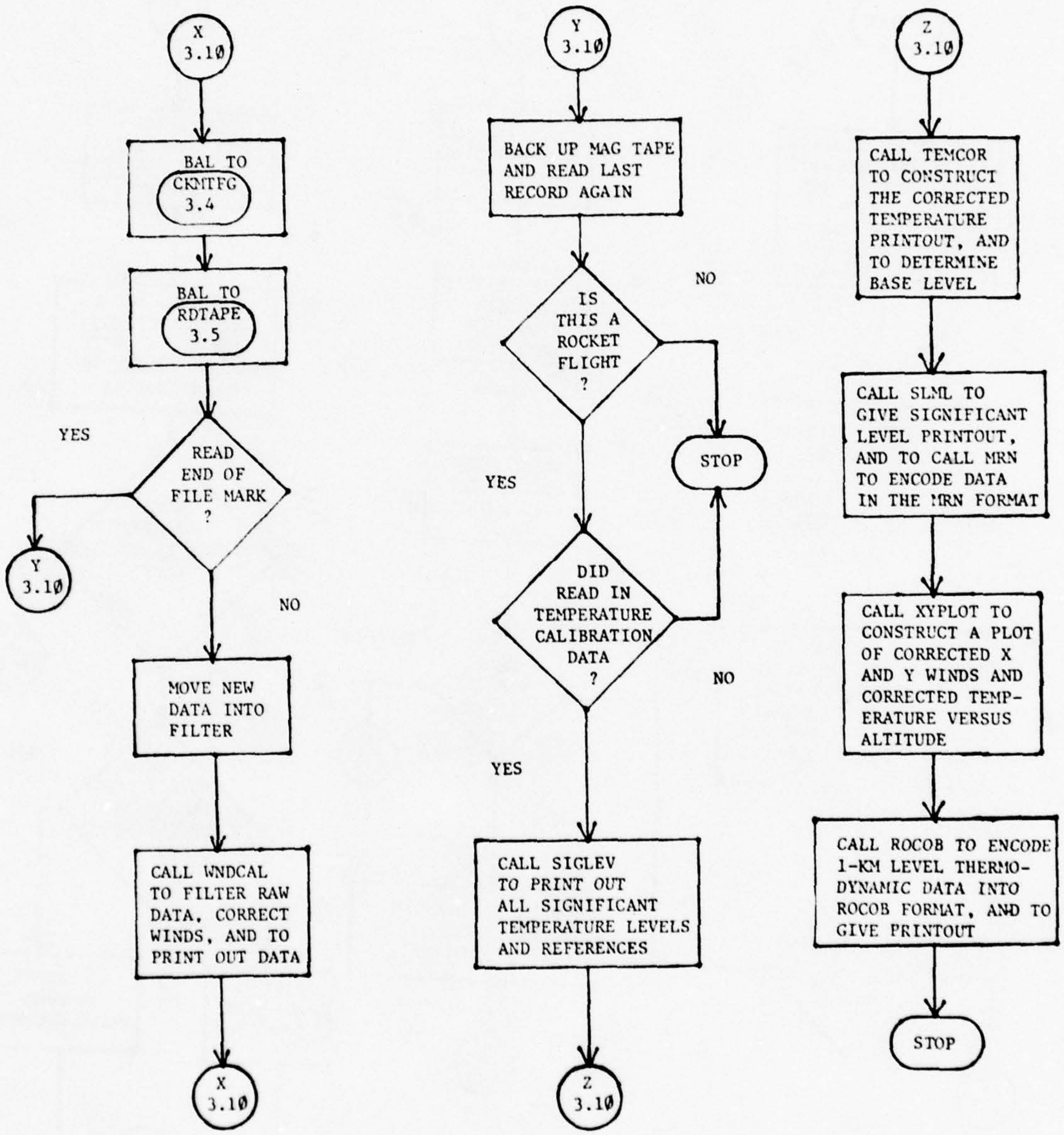
FLOWCHART FOR MAIN PROGRAM - 3.8



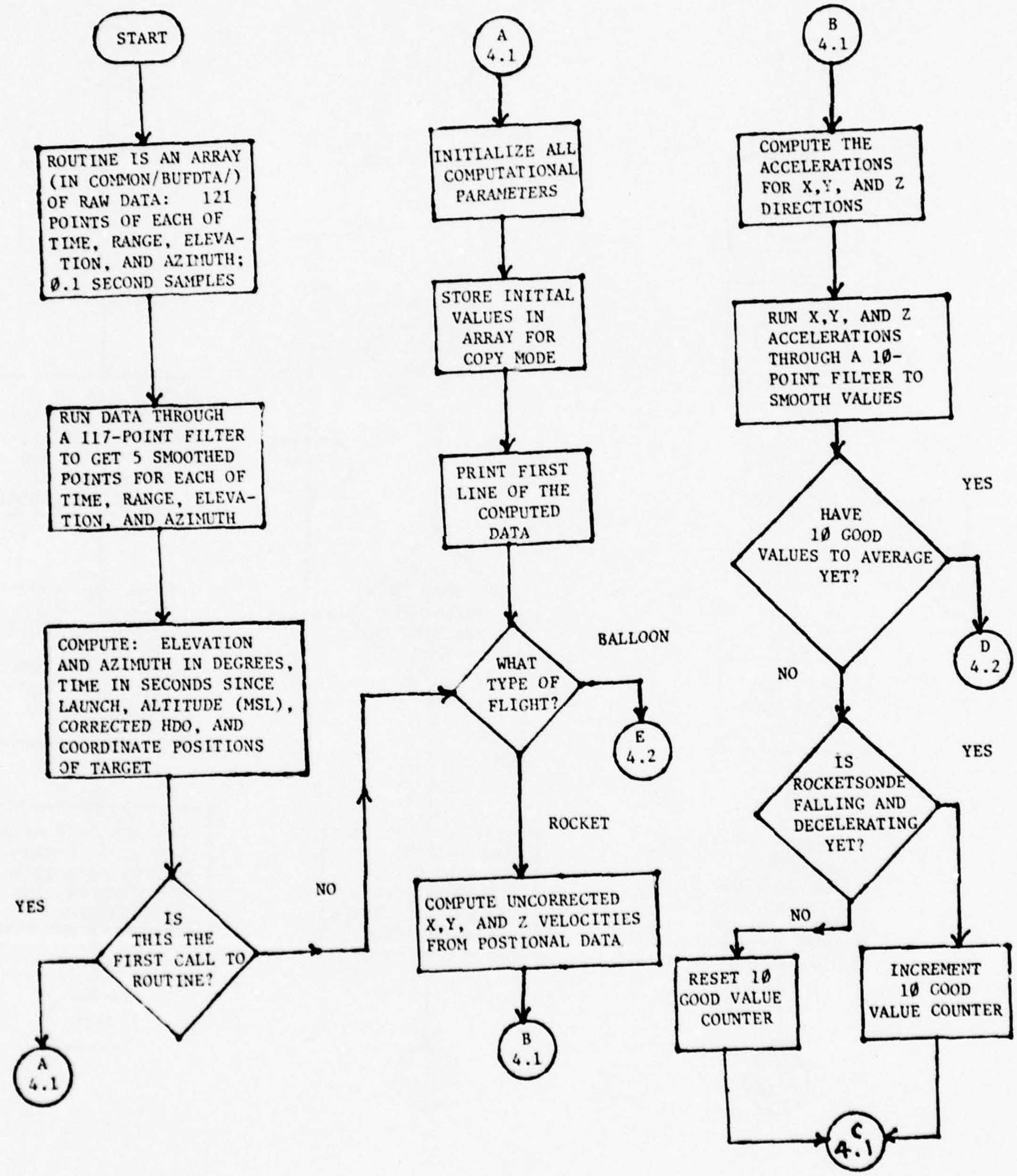
FLOWCHART FOR MAIN PROGRAM - 3.9



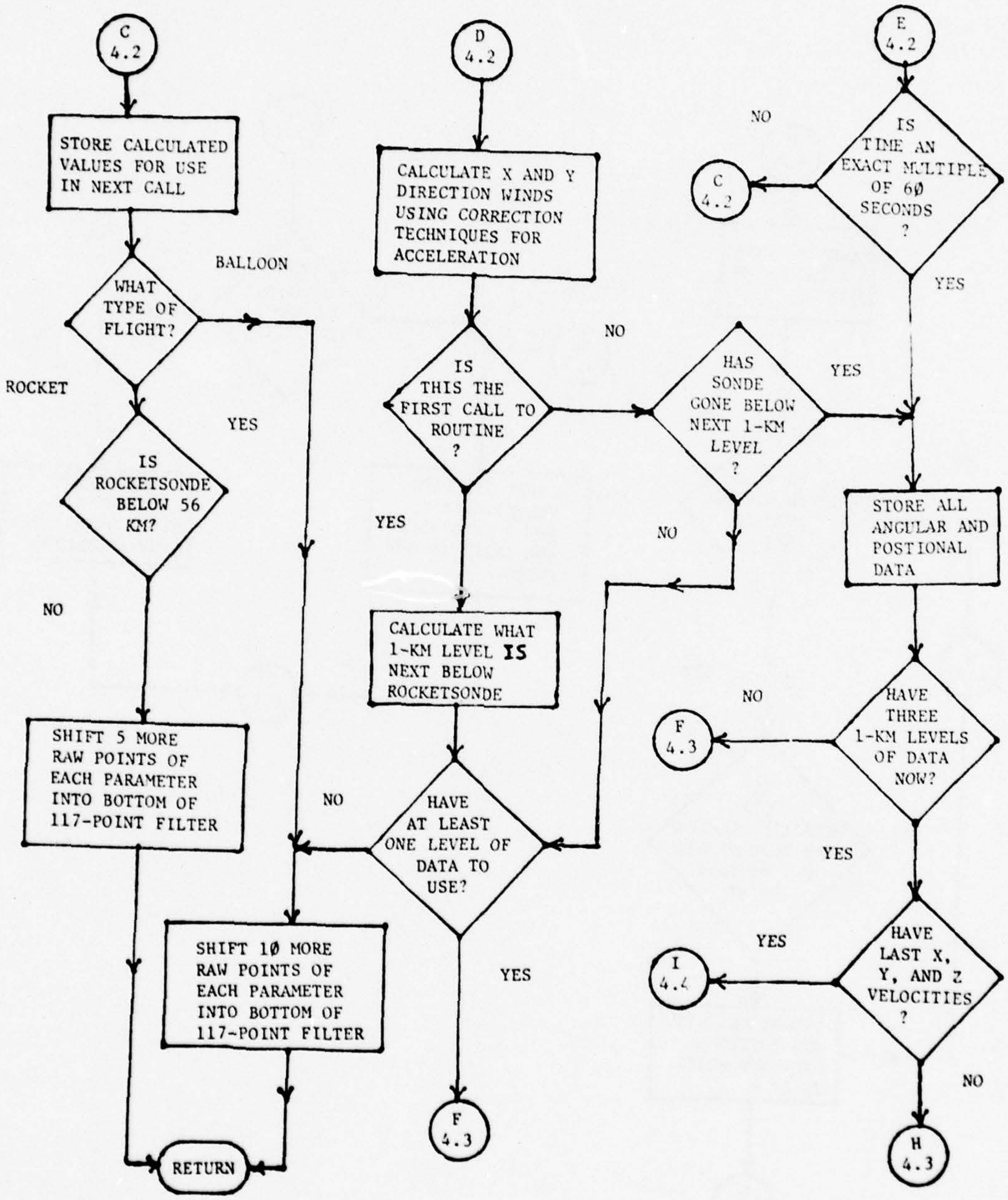
FLOWCHART FOR MAIN PROGRAM - 3.1Ø



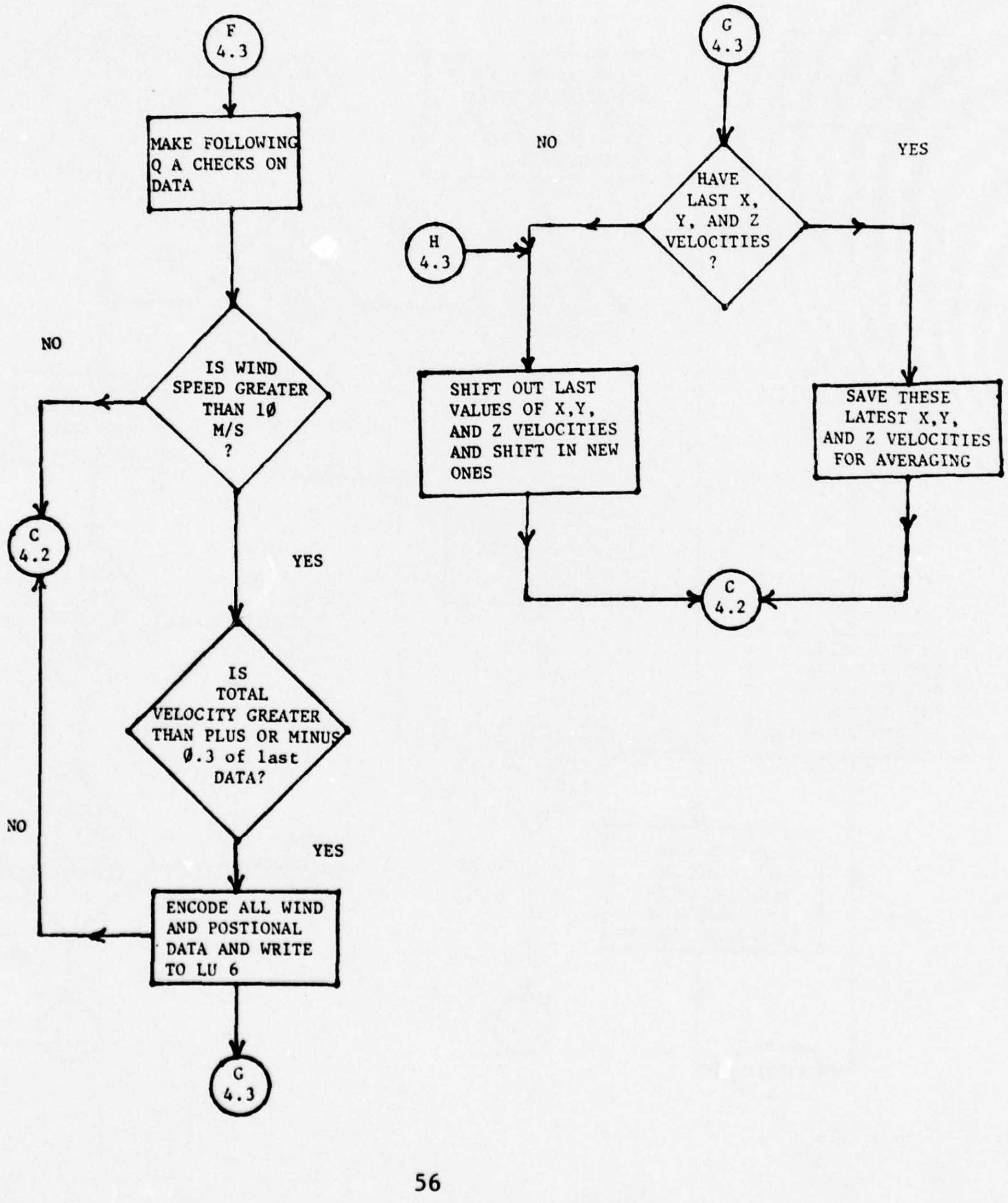
FLOWCHART FOR SUBROUTINE WNCAL - 4.1



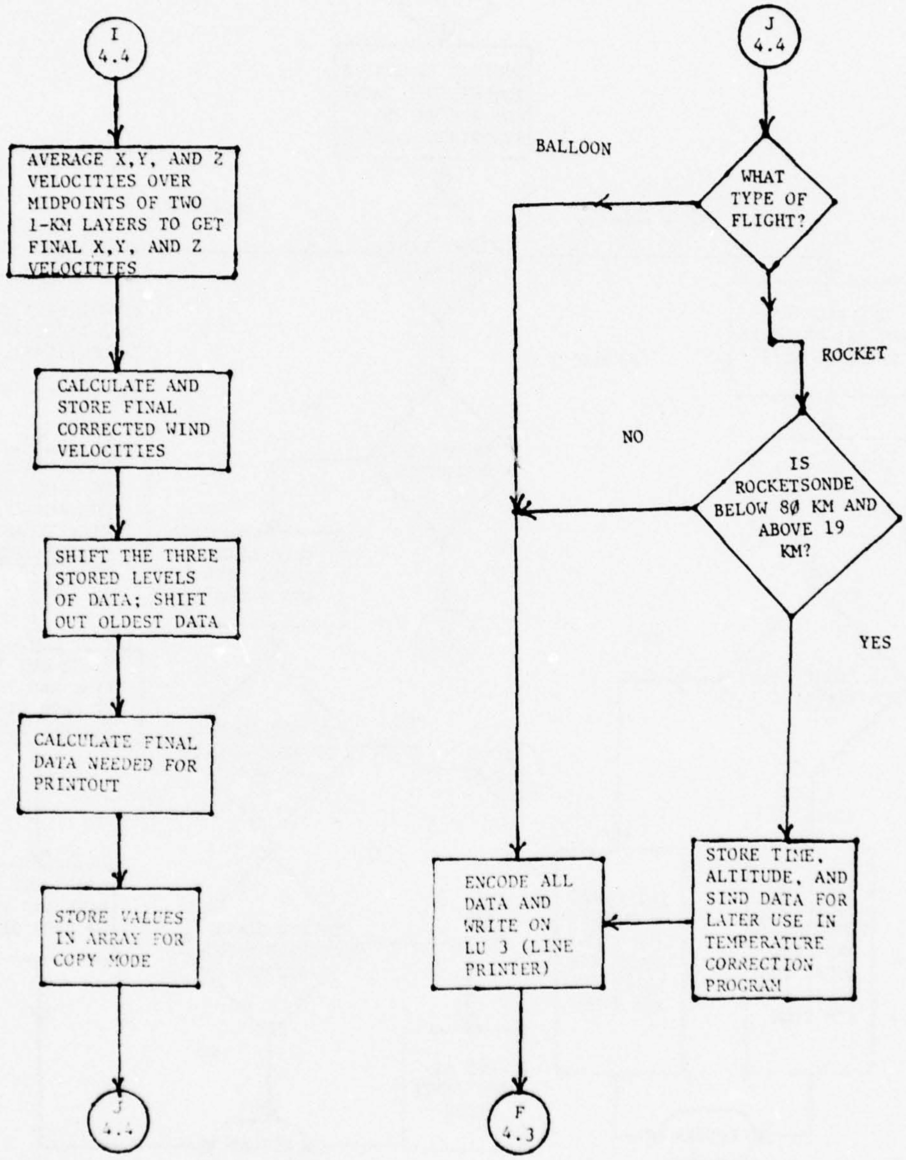
FLOWCHART FOR SUBROUTINE WNDCAL - 4.2



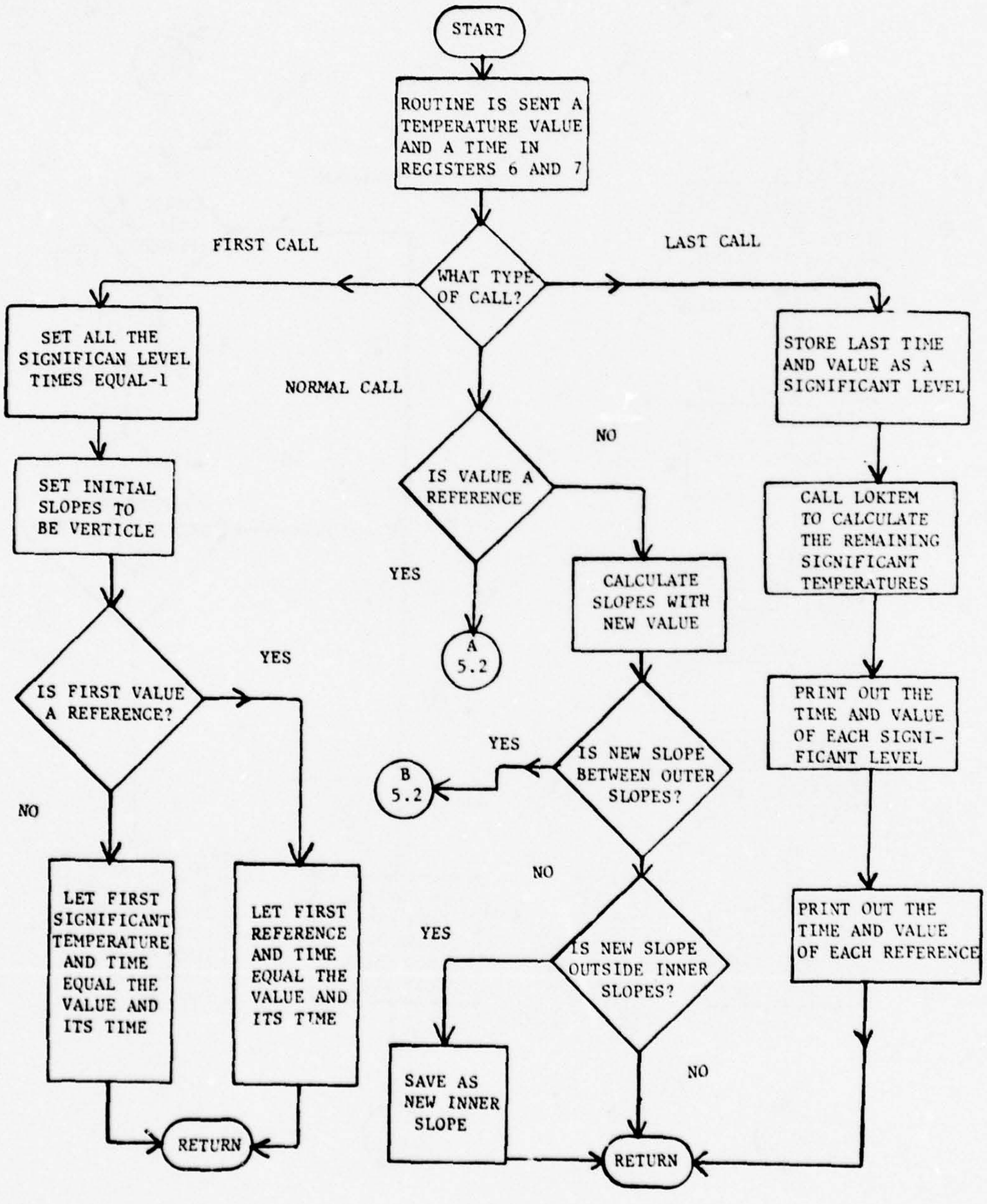
FLOWCHART FOR SUBROUTINE WNDCAL - 4.3



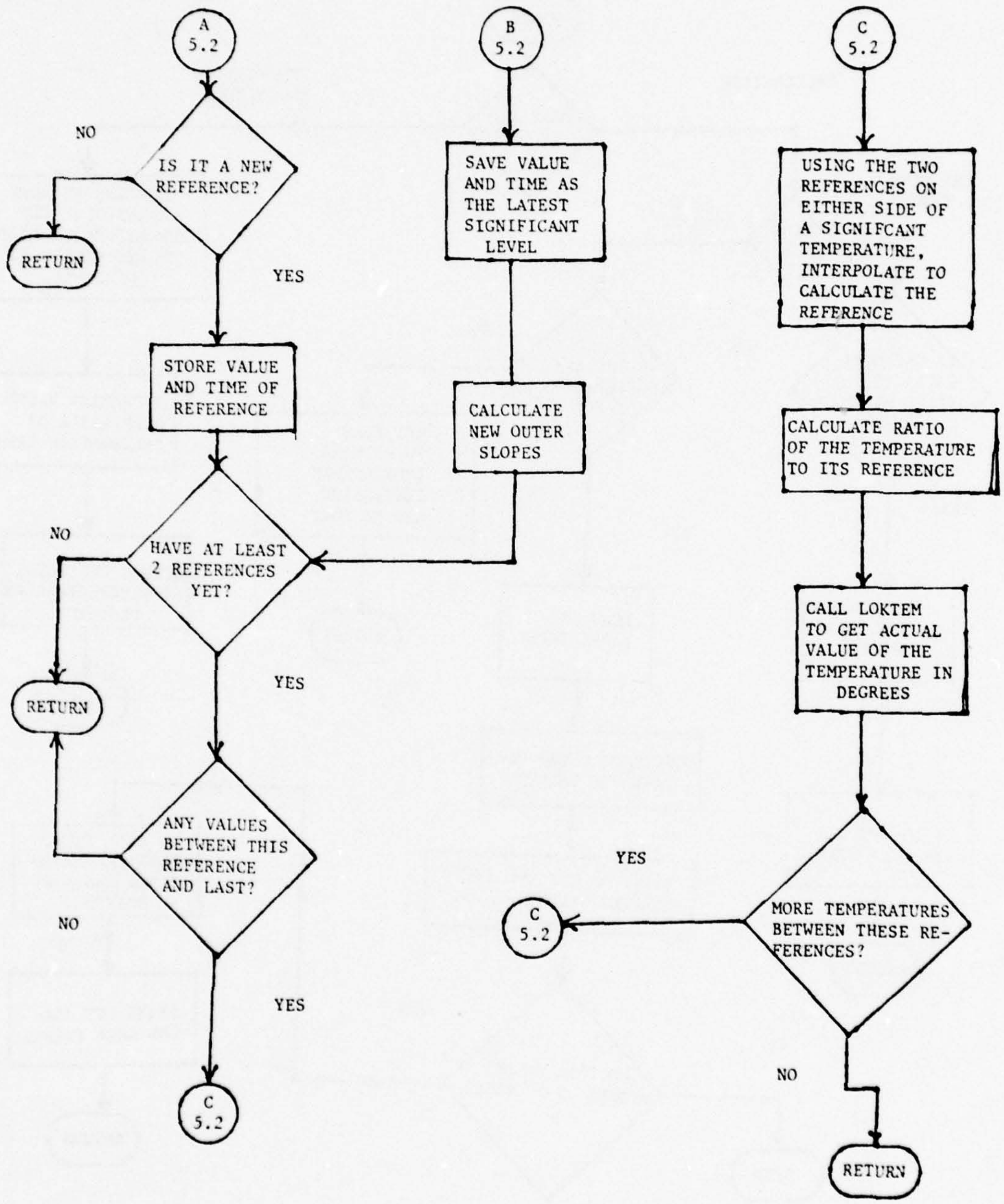
FLOWCHART FOR SUBROUTINE WNDCAL - 4.4



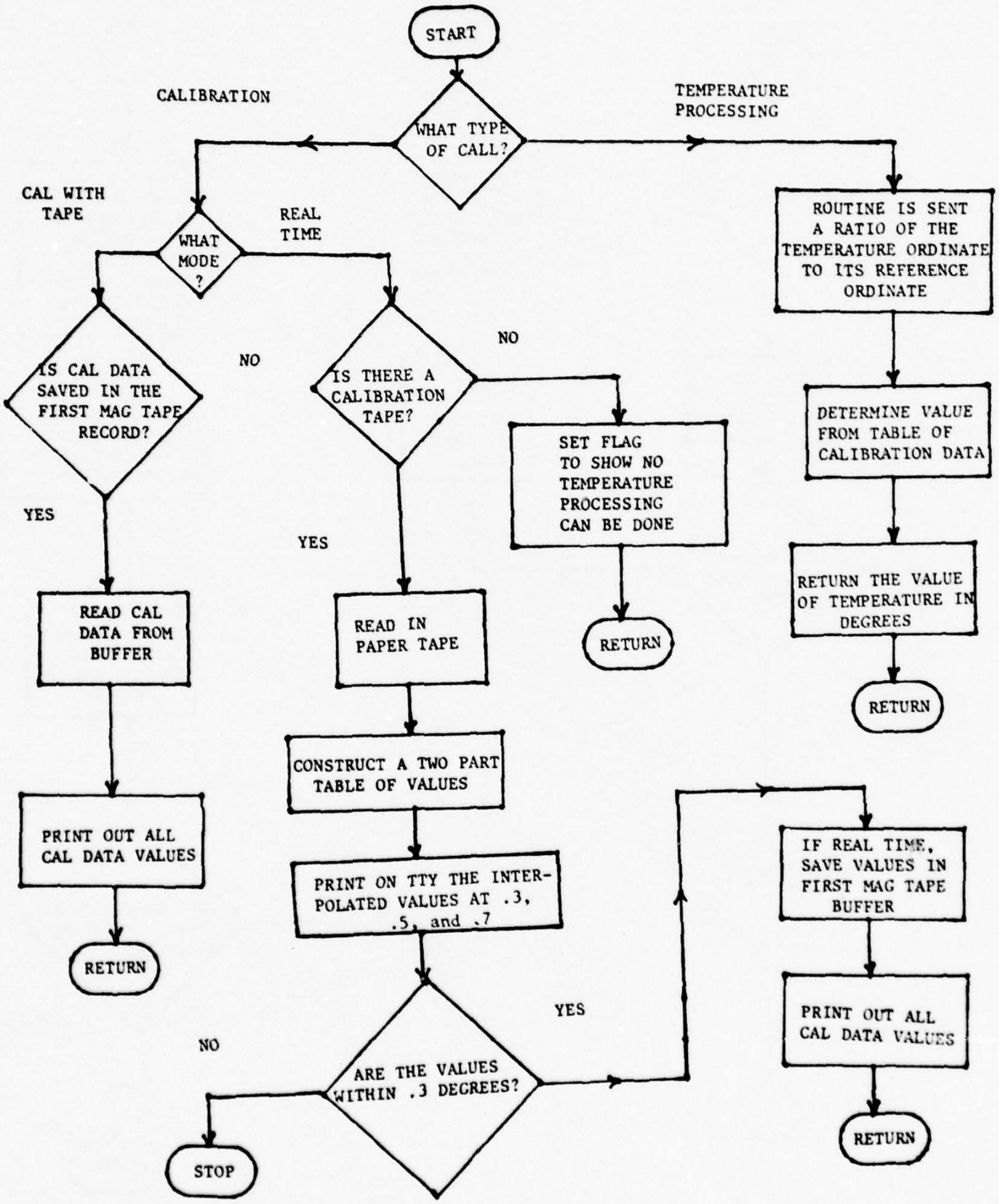
FLOWCHART FOR SUBROUTINE SIGLEV-5.1



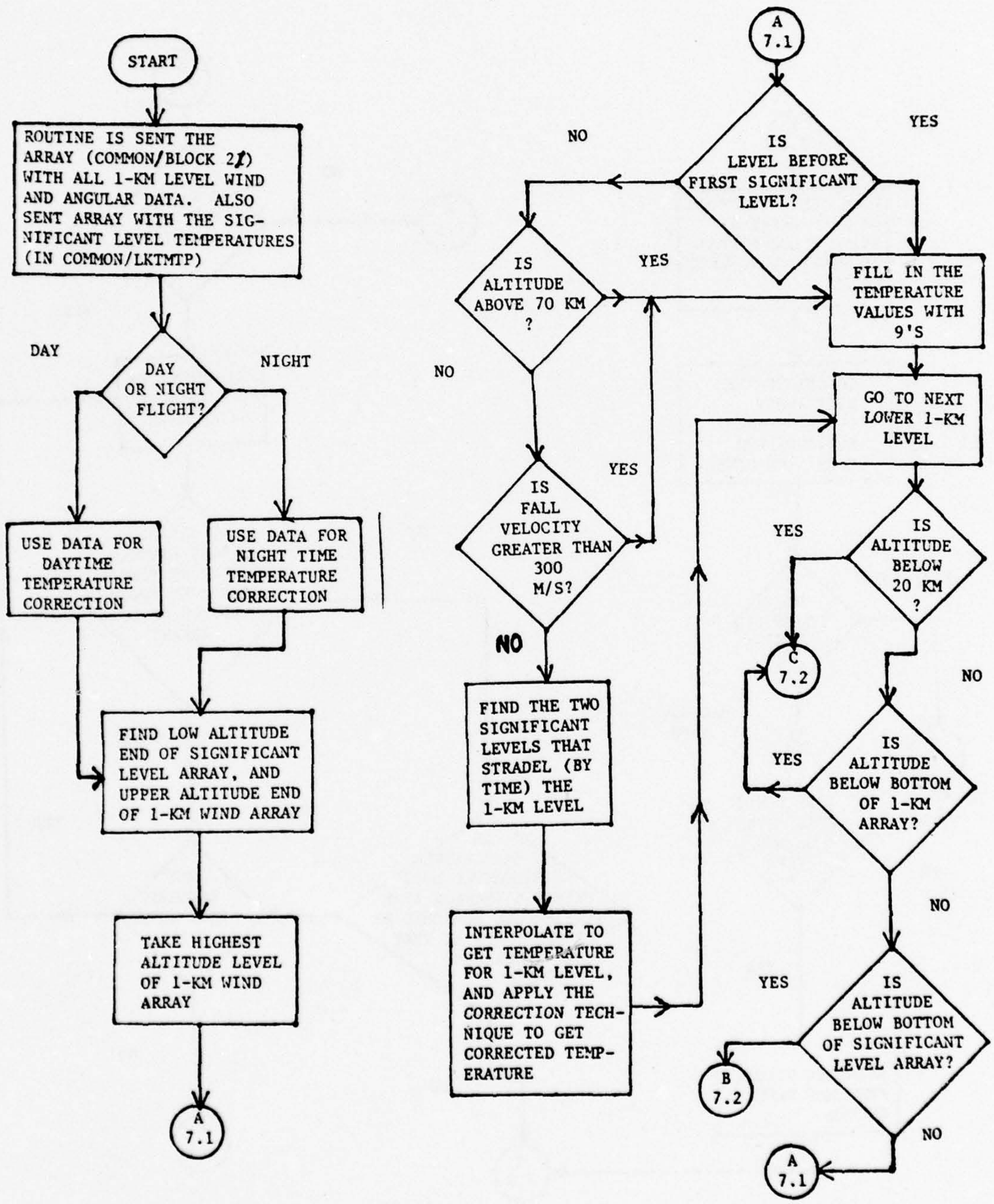
FLOWCHART FOR SUBROUTINE SIGLEV-5.2



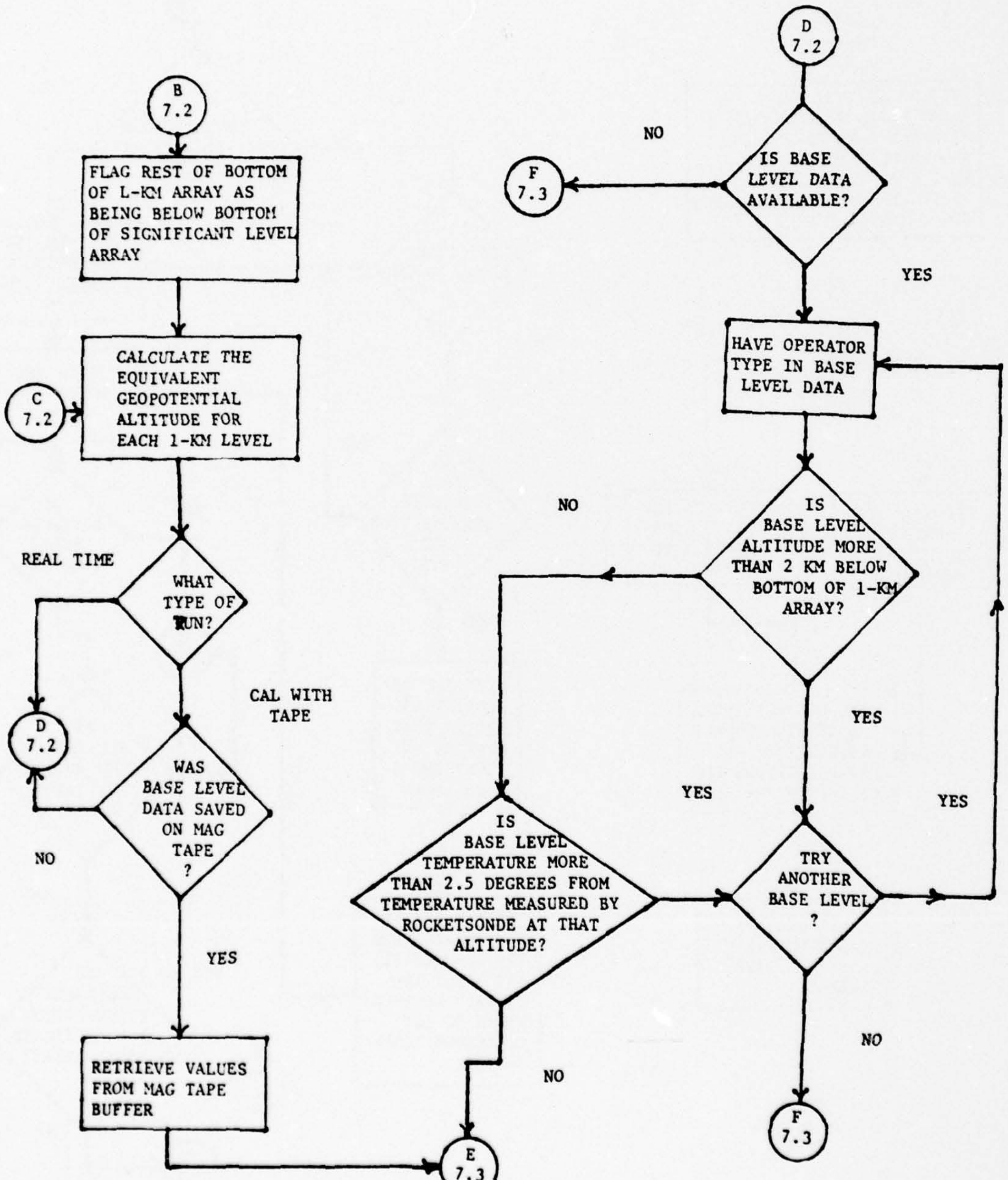
FLOWCHART FOR SUBROUTINE LOKTEM-6.1



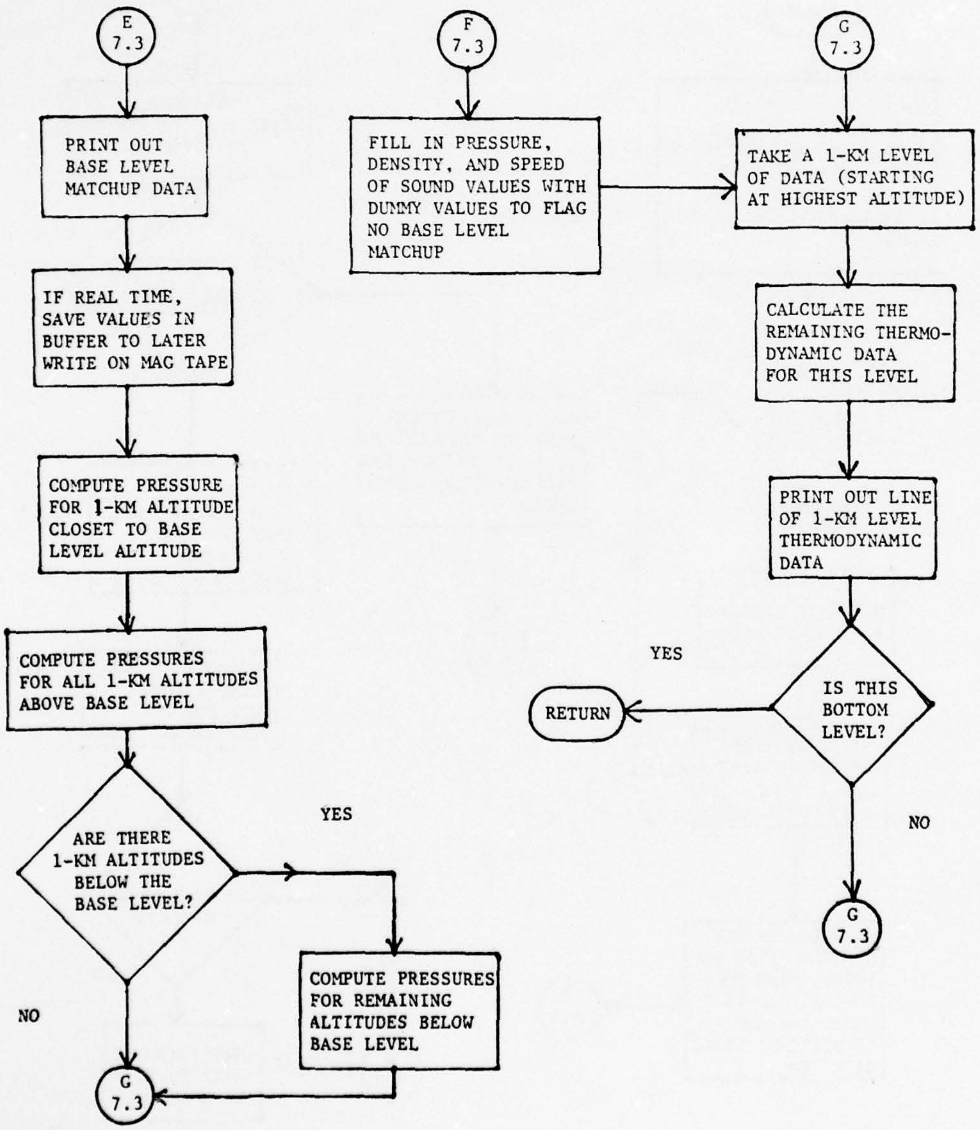
FLOWCHART FOR SUBROUTINE TEMCOR -7.1



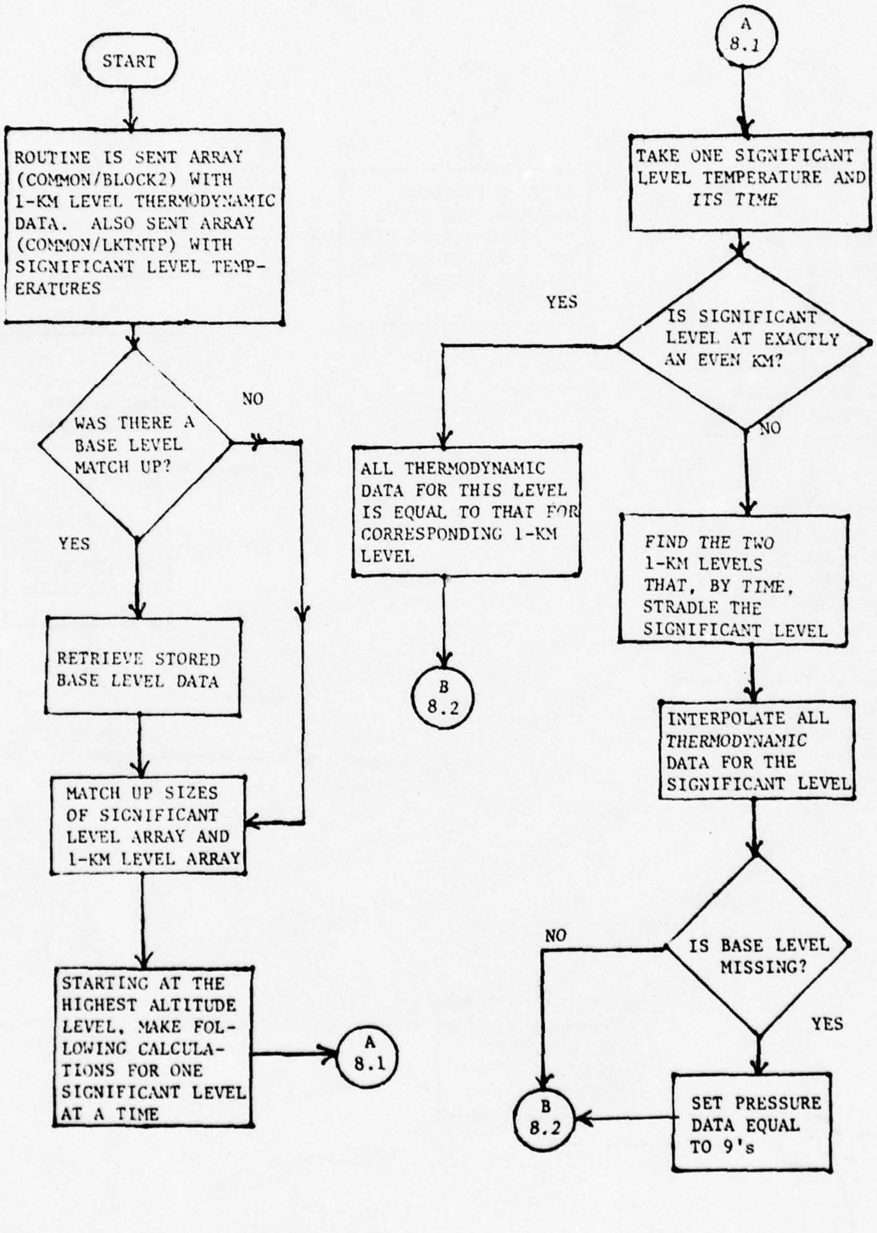
FLOWCHART FOR SUBROUTINE TEMCOR - 7.2



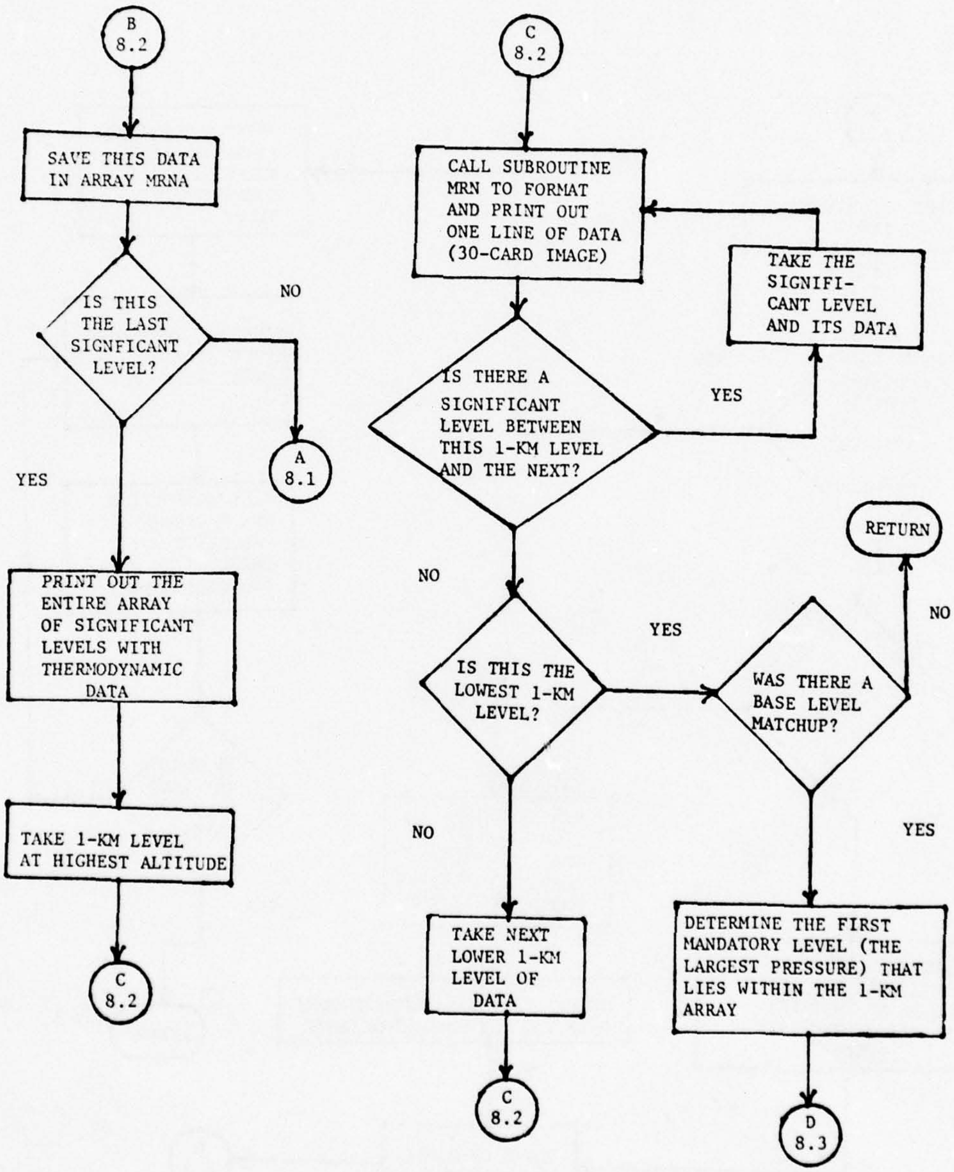
FLOWCHART FOR SUBROUTINE TEMCOR-7.3



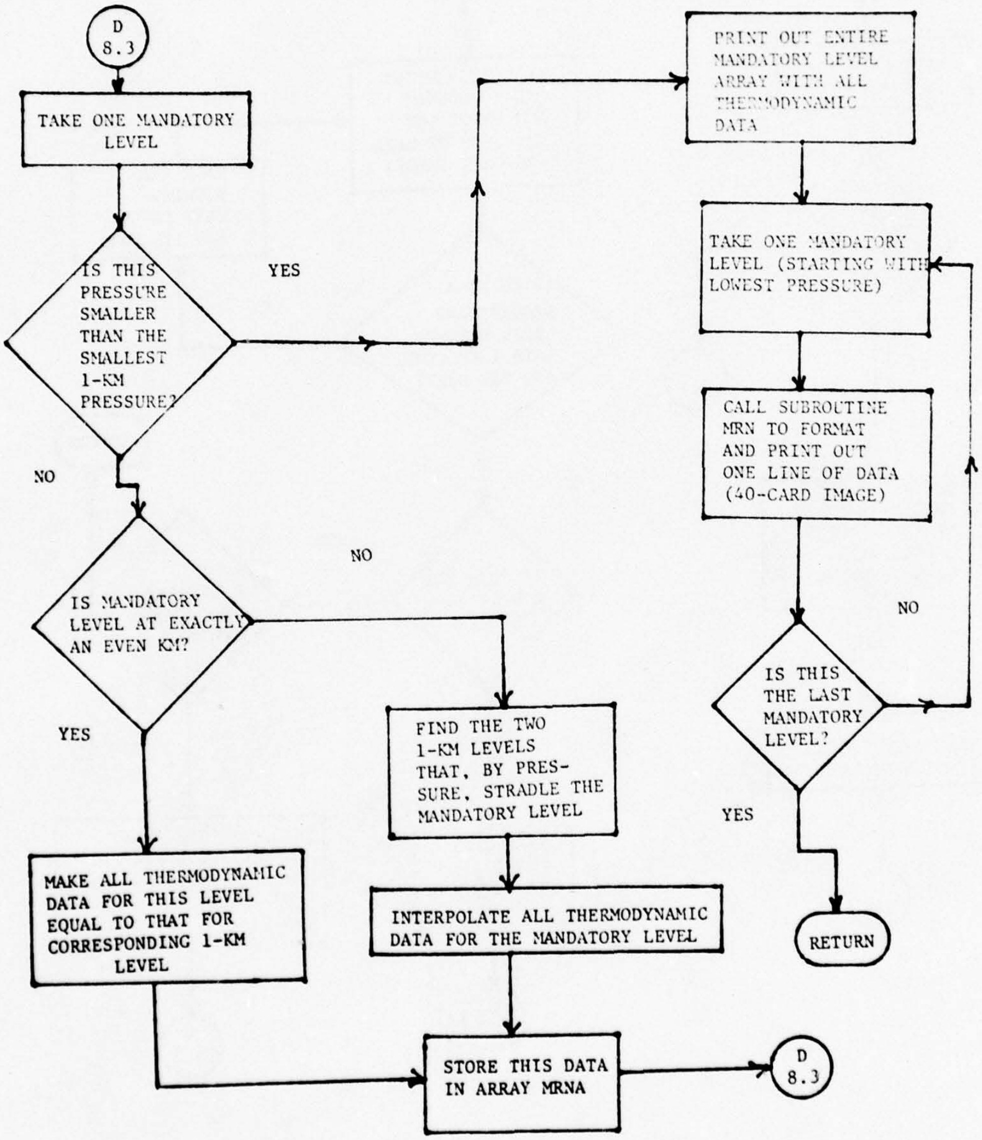
FLOWCHART FOR SUBROUTINE SLML-8.1



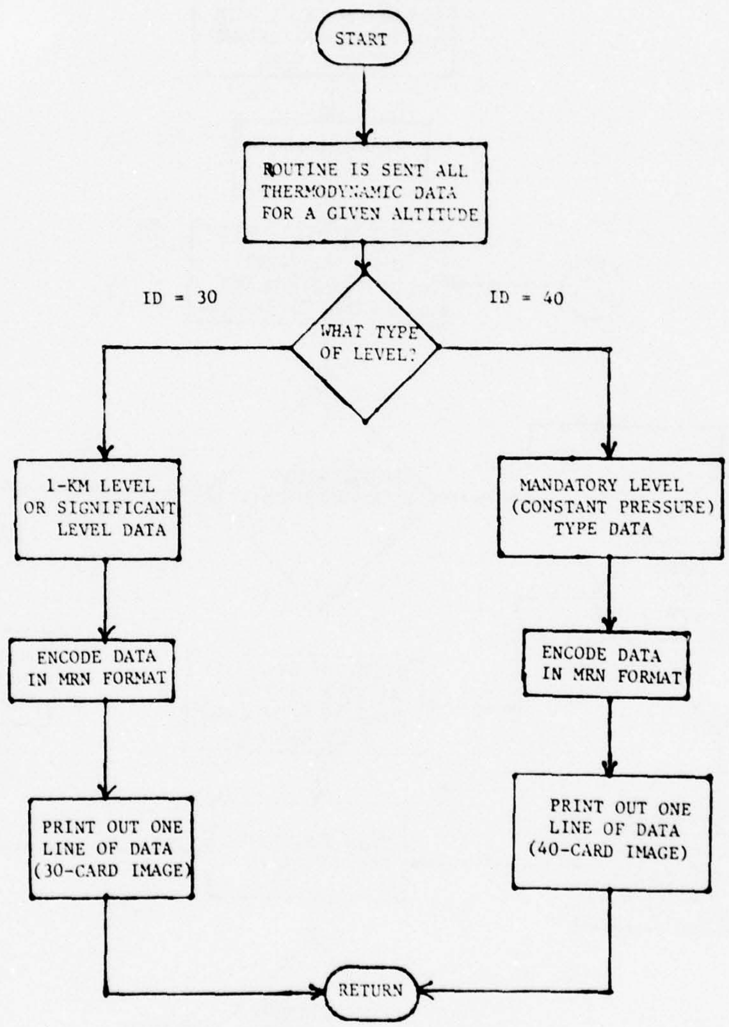
FLOWCHART FOR SUBROUTINE SLML-8.2



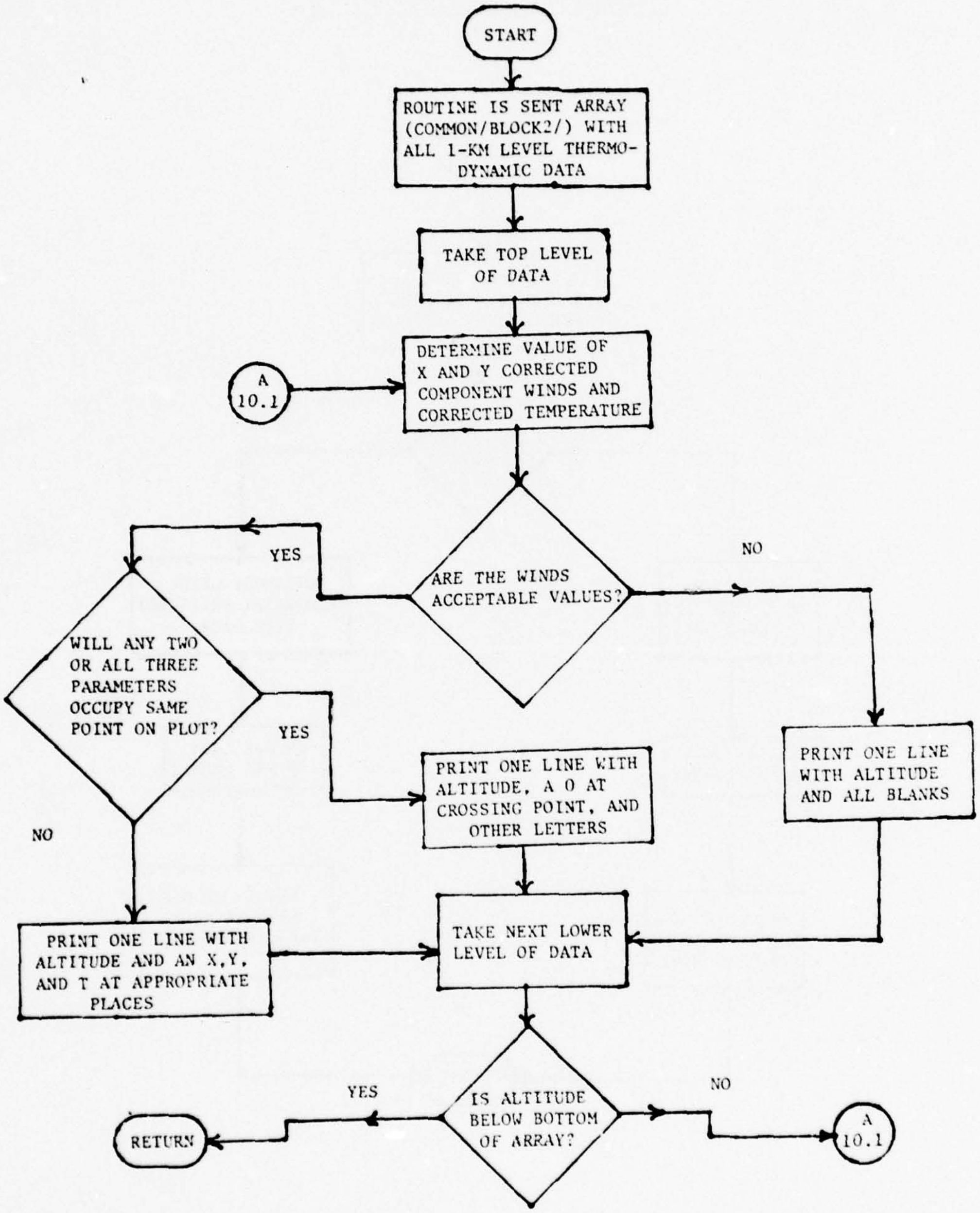
FLOWCHART FOR SUBROUTINE SLML-8.3



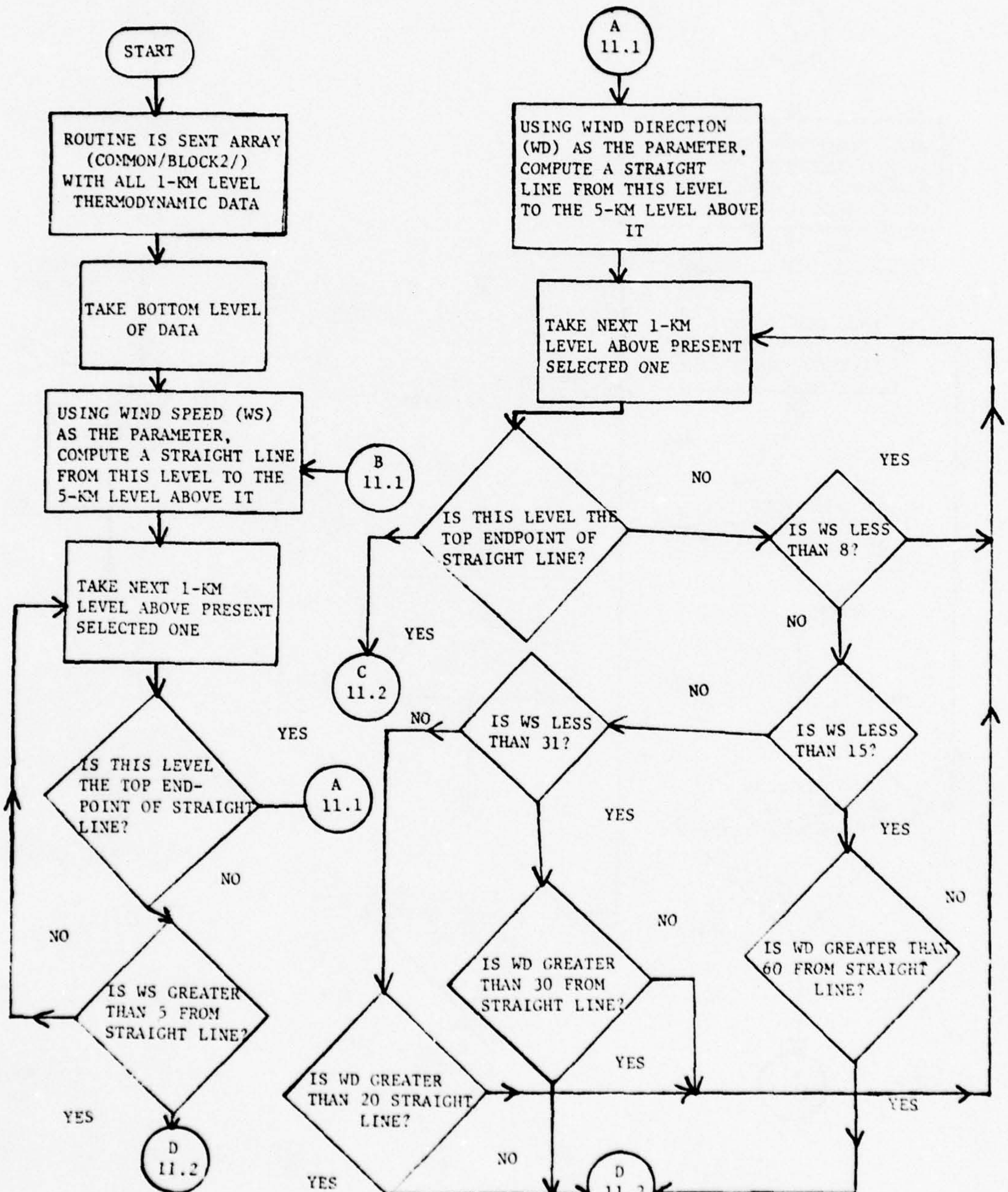
FLOWCHART FOR SUBROUTINE MRN - 9.1



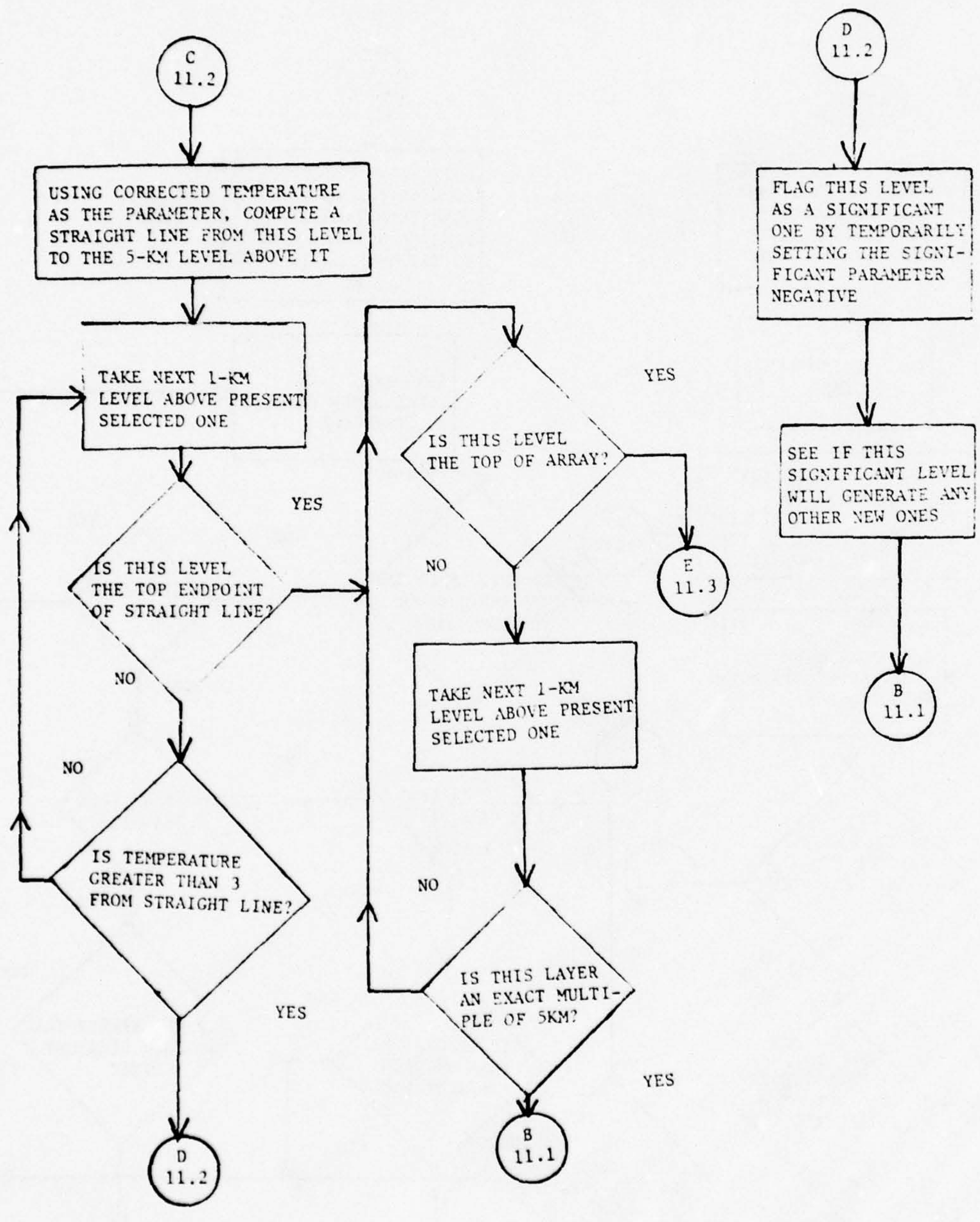
FLOWCHART FOR SUBROUTINE XYPLOT-10.1



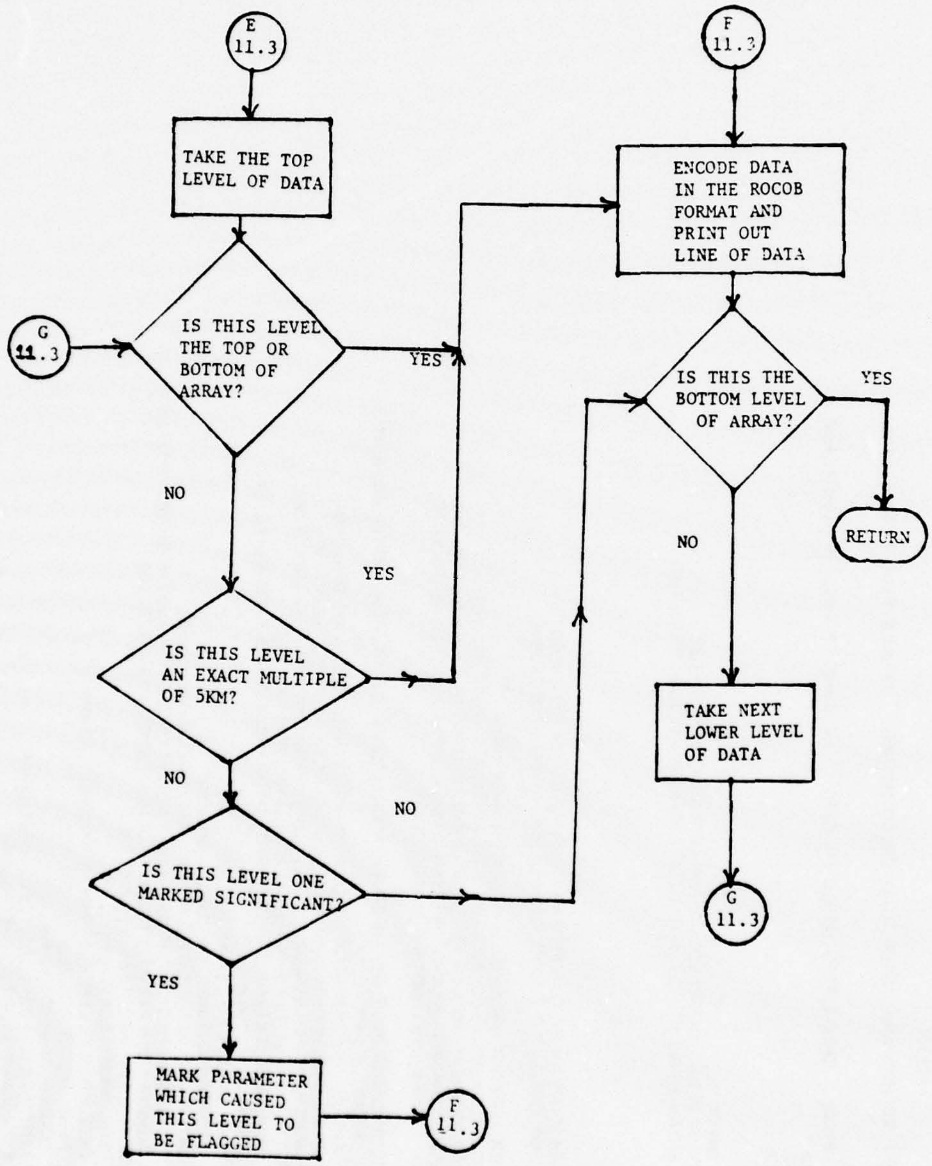
FLOWCHART FOR SUBROUTINE ROCOB-11.1



FLOWCHART FOR SUBROUTINE ROCOB-11.2



FLOWCHART FOR SUBROUTINE ROCOB-11.3



V PROGRAM LISTINGS

MAIN

```
C     MET ROCKET AND  SOND WIND CALCULATIONS   SOURCE FILE LKGMDP.FOR
$ASSM
 SCRAT
NHTTR  PROG   CALCULATES WINDS OR COPY   16 JUNE 77 R02  S LKGMDP.FOR
  CROSS
  NOSOZ
  NORX3
 NLSTC
 NLIST
        $ARGT 32
  ENTRY INITIAL,INTRP,LNC,WRTL6,WRTL3,ELOFFS,AZOFFS
      ENTRY PRAM,MTPTST,ENCDDISP,FGWNDCAL,T,IGAS,PRINTHED
*
*
*
*
$FORT
C      THIS PROGRAM COPIES PREVIOUS DATA CALCULATIONS
C       OR CALCULATES WINDS FOR ROCKETS OR RADIOSONDS
C
C  INDEX CURRENT CELL TO STORE NEXT RAW DATA
C  MINDX CURRENT CELL TO GET DATA FROM(EXT USER UPDATE)
C  METDX CURRENT NUMBER OF STORED CALCULATED MET DATA(USER SET)
C  LMETDX CURRENT NUMBER OF STORED CALCULATED MET DATA IN THIS
C           TAPE  RECORD(SET INTERNALLY)
C
C  NEXT 8 CARDS MUST BE IN ORDER ROCKT,TIMEL,TSTAR,FLAG,RTT,RSHEAR,MET,
C         RANGE,ELERR
      COMMON /BUFDTA/ROCKT,TSTAR,TLOCK,TIMEL,MN,DY,YR,OPR,OPQ
     1,FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX,RTT,RZZ,RZF,RVY,
     2 RVX,RVD,RVZ,RSHEAR,RTIMM,RSMOT9,RSMOT8,RVS,MET,TIM1,TIM2
     3,TIM3,RANGE,ELEV,AZMU,ELERR,AZERR,METRAW,LTRAW
      COMMON/BLOCK1/AA,IFRST,IPRNT,IRAN,KA1,KYY,BUFF,EB,BUFF1,EB1
     1,IBFULL,JC,IDSP
      REAL ROCKT(3),TSTAR,TLOCK
       INTEGER*2 TIMEL,MN,DY,YR,OPR,OPQ
      INTEGER*2 FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX
      DIMENSION RTT(10),RZZ(10),RZF(10),RVY(10),RVX(10),RVD(10),RVZ(10),
     1 RSHEAR(10),RTIMM(10),RSMOT9(10),RSMOT8(10),RVS(10)
      INTEGER*2 MET,TIM1,TIM2,TIM3
      REAL RANGE,ELEV,AZMU
      INTEGER*2 ELERR,AZERR,METRAW(1187),LTRAW
       REAL MONTH(12),NSH(2),EWH(2)
       DOUBLE PRECISION POS(2)
       INTEGER*2 EOF,I,JC,ICAL,IBFUL,IDSP
      INTEGER*2 ITM(4),IPRESS(180)
      DIMENSION C(3),RAW(4,121)
      REAL OFS,TM,TLORD
      REAL AZADDL,ELADDL,C16BDG,ELOFFS,AZOFFS,ELNHR,AZNHR,C17BDG
      INTEGER*4 NHREL,NHRAZ,RGNHR
```

```
      INTEGER*2 ADLDEL,ADLDAZ,ORDV
      REAL LN1(7),LN2(9),LN3(8),LN4(8),LN5(8),LN6(8),LN7(8),LN8(8)
      REAL LN9(8),LNA(8),LNB(8),LNC(8),LND(8),LNE(8),LNF(8),LN16(8)
      REAL LN17(8)
      INTEGER*2  KA4,L,FMET,NSAM,NSUM,T,FF,ORD,TOVV,OVL,FORD,TOV
      INTEGER*4 IRAN(4,121)
      INTEGER*4 EL,AZ,RG,ELP,AZP,RGP
      INTEGER*2 IHD,I1SIG,NODF,IGAS
      DIMENSION  BUFF(33),BUFF1(33)
      INTEGER*2 J,KA1,KYY,EB,EB1
       INTEGER*2 IOF,IND,AA,IFRST,IDISP,IPRLP
      INTEGER*2 ITH,IH,ITMS,IM,ITS,IS,IPS
      EQUIVALENCE(EL,ELF),(AZ,AZF),(RG,RGF),(IRAN(1,1),RAN(1,1))
      DATA DLOKI,DSONDE,NODF/180.,120.,0/
      DATA T,NSUM,NSAM,D360/20,6,5,360./
       DATA POS,NSH,EWH/'POSITIONUNC  VEL','+N-S-N+S','+E-W-E+W'/
$ASSM
  NLIST
$FORT
      DATA OFS,C17BDG,C16BDG,ELOFFS,AZOFFS/0.,.0027466,.00549326,0.,0./
      DATA C/3600.,60.,1./
      DATA LN1/'  EL-DEG  AZ-DEG  SK-MTR TEL'/
      DATA LN2/'
      DATA LN3/'                                  '/
      DATA LN4/'                                  '/
      DATA LN5/'                                  '/
      DATA LN6/'                                  '/
      DATA LN7/'                                  '/
      DATA LN8/'                                  '/
      DATA LN9/'                                  '/
      DATA LNA/'                                  '/
      DATA LNB/'                                  '/
      DATA LNC/'                                  '/
      DATA LND/'                                  '/
      DATA LNE/'                                  '/
      DATA LNF/'                                  '/
      DATA LN16/'                                  '/
      DATA SLAT,SLONG,SHLT/65.117,147.459,412./
C WS  DATA SLAT,SLONG,SHLT/32.412,106.494,1362./
      DATA C1,GG,RE/.0174533,9.822888,6370233./
C WS   DATA C1,GG,RE/.0174533,9.795168,6347250./
      DATA MONTH/4H JAN,4H FEB,4H MAR,4H APR,4H MAY,4H JUN,4H JUL,4H AUG
     1,4H SEP,4H OCT,4H NOV,4H DEC/
C
      GO TO 5
$ASSM
  NLIST
*   START REAL TIME DATA COLLECTION
*
*    INITIALIZE FOR RADAR DATA TRANSFER
```

```
******
INITIAL  LHI     6,INTRP
  STH  6,X'1EA'
   LHI  6,DISPINT
   STH  6,X'1E6'  SET UP DISP INTERUPT ADDRES
  LHI  6,X'60'
  LHI  5,X'8D'  RANGE CARD ADR
  OCR  5,6     SET CA3 FOR INTERUPT
  LHI  5,X'2203'  SET UP WRITE TO
  STH  5,WRTL3   LINE PR: PROC UNC
  LHI  5,X'2206'
  STH  5,WRTL6
          LIS    5,0
   STH  5,IBFUL   RESET MAG TAPE PRINT CODE
          STH    5,INDEX
          STH    5,MINDX
 BFCR  0,15       RETURN
INTRP  ST  0,REG0SV
  ST  1,REG1SV
SAVREG  STE  0,FLTSV
   STM  0,REGSF
   LH  4,INDEX
ANGLE       LHI  6,X'8B'   GT ANGLE ADR
            LHI  5,X'E8'   '  COMD  W3
            OCR  6,5
            RHR  6,2       READ MSB BIT OF EL(BIT 15 OF C)
            SIS  5,4        AND EL ERROR VOLTAGE
           OCR  6,5
            RHR  6,3       READ EL LSBS
            SRHA  2,1
            BFFS  8,4        MSB =1
            AI  3,Y'10000' YES
            NHI  2,X'FFF'
            SHI  2,X'800'  -BINARY ERROR(EL)
            STH  2,ELERR(4)
            ST  3,ELEV(4)
  FLR  0,3
  MF  0,C17BDG
  STE  0,ELNHR
            SIS  5,2
            OCR  6,5
            RHR  6,2       READ MSB AZ
            SIS  5,1
            OCR  6,5
            RHR  6,3       READ AS MSB
            SRHA  2,1
            BFFS  8,4       MSB=1
            AI 3,Y'10000'  YES
            NHI  2,X'FFF'
            SHI  2,X'800'  -BINARY ERR(AZ)
```

```
              STH  2,AZERR(4)
              ST   3,AZMU(4)
   FLR  0,3
   MF   0,C17BDG
   STE  0,AZNHR
 GTIME        LHI  5,X'E8'
              LHI  6,X'8C'    TIME ADR
              OCR  6,5
             RH  6,TIM1(4)  (1YR,100D,10D,D)
              SIS  5,4
              OCR  6,5
             RH  6,TIM2(4)  (10H,H,10M,M)
              SIS  5,2
              OCR  6,5
             RH  6,TIM3(4)  (10S,S,PS,F)
              SIS  5,1
              OCR  6,5
              RHR  6,2       4TH MET
              NHI  2,X'FFF'   12 BITS
              SHI  2,X'800'   -MET
              STH  2,MET(4)
    LIS   3,0
    SR   3,2  +MT
    FLR  0,3
    DE  0,D20
    STE  0,TLORD
 GRANGE       LHI  5,X'E8'
              LHI  6,X'8D'    RANGE ADR
              OCR  6,5
              RHR  6,3       100 (MSBS RANGE)
             SIS  5,4
             OCR  6,5
             RHR  6,5   1 LSBS RANGE
             SLLS 3,1
              TI   5,X'8000'
              BFFS 3,2
              AIS 3,1        3 MUST BE ODD
              NHI  5,X'7F'
              M    2,D100    3=100*3
              AR  3,5       3=NANOSEC
              M    2,CNSTM   CONV NS TO METERS
              ST   2,RANGF(4)
    ST  2,RGNHR
 $FORT
 4      CONTINUE
 $ASSM
  LCS 5,1
  AHM 5,FMET  MET FLAG SET
  BTC  2,THRU   NO
  LH 5,NSAM   TAKE DATA EVERY NSAM TIMES  YES
```

```
 STH 5,FMET  RESET MET FLAG
**************START OF FILTER ROUTINE CONTROL*********
*
*  FF=1  THIS SAMP PART OF PREV STRING
*  FF=0  BAD DATA POINT
*  FF=-1 LAST TWO SAMPLES FORM NEW STRING
*
 LIS  6,0
 SH  6,MFT(4)
 BAL  14,SAMP
 LH 5,FF  FF=-1 OR 1
 BTC 3,FF1  YES
 BFC 0,FF0  NO
TEST  LH 7,FF  FF=-1
 BFC 1,TEST1  NO
 AHR 5,6 YES
 STH 5,ORDV
 LIS  6,2
  SH  6,NSUM
 STH 6,NSA NSA =2-NSUM
  LIS 6,0
 STH 6,NST NST  NE 1
 BFC 0,THRU
TEST1  LH 4,NST FF=1,NST.NE.0?
 BTC 3,THRU   YES
 AHM 6,ORDV   NO
 LIS 5,1
 AHM 5,NSA  NSA=0
 BTC 3,THRU NO
 STH 5,FORD FORD=1
 STH 5,NST NST=1
  LH  5,ORDV
  LH  6,ORDV
   LH  4,OPR   OPR =1?
   BFC  2,TEST1A   NO
   LIS   4,0   YES
   STH  4,P2   RESET
   STH   4,P1  ORD  FLAG
   BFC  0,TEST3
TEST1A   SH  6,OVL
  BFC  1,TEST2
  XHI  6,X'FFFF'
  AIS  6,1
TEST2  SH  6,TOV  ABS(ORDV-OVL)<TOV
  BFC  1,TEST3  NO
  LIS  6,0  YES
  STH  6,FORD
  LH  5,OVL
TEST3  STH  5,OVL
*******END CONTROL FILTER*****
```

```
THRU  LH 5,JC
  CHI 5,0
  BTC 2,#P875
   LH   4,INDEX
     AHI  4,H'24'
    STH  4,INDEX
    CLHI   4,H'2400'
    BTC  1,STRET
    LIS   5,0
    STH   5,INDEX
    LH   6,METDX
    STH   6,LMETDX
    STH   5,MFTDX
    LIS  5,0
    CH  5,IDISP  DISP ONLY
    BFC  3,STRET   YES
  LIS  5,1   SET MAG TAPE WRITE FLAG
   STH  5,IBFUL
STRET  LIS  5,1
   STH  5,IDSP
STR  LHI  5,X'8D'
  LHI  6,X'60'
  OCR  5,6  SET UP RGN INTERUPT
  LM  0,REGSF
  LE  0,FLTSV
  LPSW   REG0SV  RETURN
  ALIGN  8
REG0SV  DS  4
REG1SV  DS  4
STHTINTR  DC  Y'62F0'
    DC   A(SAVREG)
D20  DC  E'20'
REGSF  DAS  16
FLTSV  DS  4
**********START OF FILTER**********
FF0   STH  6,P4
      LH   5,P3
      BAL  14,ABSO   ABS(P4-P3)<T
      BTC 1,FF01   YES
      STH  6,P3   NO
      BFC  0,FF11
FF01  LH   5,P2      YES
      BAL  14,ABSO  ABS(P4-P2)<T
      BFC  1,FF01A      NO
      LIS   4,1      YES
      BFC  0,FF01B
FF01A LH    5,P3
      STH   5,P1    P1=P3
      LCS   4,1
FF01B STH   4,FF    FF=-1 OR 1
```

```
       STH   6,P2     P2=P4
       BFC   0,TEST
*      SAMPLE IN R6
*          VALID  RET (14) +4
SAMP  CH  6,ORS  SAMPLE< 5 ORD
 BTC 1,THRU  YES
       CH    6,FS  SAMPLE > FS
 BTC 2,THRU YES
       BFCR  0,14 GOOD SAMPLE RET
*     ENTER HERE FOR FF=1 OR -1,SAMPLE IN R6
FF1   STH 6,P3
FF11  LH 5,P2         R6=P3
       BAL 14,ABSO    ABS(P2-P3)<T
       BFC 1,FF13      NO
FF12 LH 5,P2  YES
       STH 5,P1        P1=P2
       STH 6,P2        P2=P3
       LIS 5,1
       STH 5,FF        FF=1
 BFC 0,TEST
FF13  LH  5,P1
       BAL 14,ABSO    ABS(P1-P3)<T
       BTC  1,FF12  YES
       LIS  4,0   NO
       STH  4,FF      FF=0
 BFC 0,THRU
*      R5,R6 CONTAIN VALUES
*     IF ABS(R5-R6)<T RET COND CODE=1 R6 SAMF
ABSO  STH   6,SV
       SH    5,SV    (5)-(6)
       BFC  1,ABSOA
       XHI   5,X'FFFF'
       AIS   5,1
ABSOA SH    5,T        ABS((5)-(6)) -T
       BFCR  0,14
  NLIST
NSA  DS  2
NST  DS  2
ORS  DC  H'40'  PASS DATA 4 ORD VALUES
FS  DC  H'2000'  FS TEST
SV  DS  2
P1  DS 2
P2  DS 2
P3  DS 2
P4  DS 2
*******END OF FILTER*******
*    *COMMON HEADING PRINT*     *
PRINTHED    ST   15,PRINTSAV
$FORT
       WRITE(3,780)ROCKT,ROUNDN,DY,MONTH(MN),YR,TIMEL,SLAT,SLONG,SALT
```

```
      IF (IGAS.EQ.1) GO TO 180
C   WRITE GEOMETRIC ALT
      WRITE(3,730)
      GO TO 190
C   WRITE GEOPOTENTIAL ALT
180   WRITE(3,735)
190   CONTINUE
$ASSM
   L   15,PRINTSAV
    BFCR   0,15
   ALIGN   4
PRINTSAV   DS    4
   NLIST
DISPINT   ST  0,DPR0SV
   ST  1,DPR1SV
  DC  Y'2000200'
DISPNX   STM  0,DREGSF
  LHI  2,X'8B'  DISP  ADR
 LHI  6,X'EC'  DISARM  HW MODE  COT4 COT5 DATA READY
  OCR  2,6  OUTPUT DATA READY
  LCS  5,1  DATA DEC
  LI  6,Y'FFFFFF80'  ADR  DEC
  LI  7,Y'FF80'  START ADR FOR MEM DISP
  LHI  3,X'1FF'  STARD DATA ADR
DTLD  LB  4,LN1(3)  LOAD DATA BYTE
  OR  4,7  ADR+DATA
  WHR  2,4  OUT BYTE&ADR
  AR  7,6  DEC  ADR
  AR  3,5  DEC  DATA ADR
  BNM  DTLD  LAST DATA
  LIS  6,0
  OCR  2,6   TURN OFF DATA READY
  LM  0,DREGSF
   LPSW  DPR0SV
  NLIST
  ALIGN 4   LU 1 IS MAG TAPE
PRAM   DC  Y'30010000'   WRITE BINARY SEQU AND PROC
   DC A(ROCKT)
PRAM2  DC  A(LTRAN)+1
 DS 4
 DS 4
TPSAVE   DS   4
DPSAVE   DS   4
MTPTST  ST  15,TPSAVE
   LIS   5,1
   CH  5,IBFUL
   BTC  3,MTSTEX
   SVC  1,PRAM  WRITE
   LIS  5,0
   STH  5,IBFUL
```

```
     L    15, TPSAVE
MTSTEX  BFCR  0,15   RETURN
ENCDDISP    EQU *
     ST    15, DPSAVE
    LIS  5,0
     CH   5, IDSP
     BFC    3, RTDISP    DISP  SET
     STH    5, IDSP    YES  ENCODE    DATA
$FORT
           ENCODE(LN2,300)ELNHR, AZNHR, RGNHR, TLORD
$ASSM
     LHI    6, X'60'
    LHI  5, X'8B'
    OCR  5,6  SET DISPLAY INTERUPT
     L     15, DPSAVE
RTDISP  BFCR    0,15    RETURN
      ALIGN  8
DPR0SV    DS  4
DPR1SV    DS  4
DPNPSW    DC  Y'72F0'
     DC  A(DISPNX)
DREGSF    DAS   16
R15S    DS  4
D100  DC    Y'64'    DEC 100
CNSTM DC    Y'265F9965'  CNSTM=(C12 M/NS)*(2**32)
      ALIGN 4
INITMET    ST    15, R15SAVE
     LIS    5,0
     STH    5, FORD
     STH    5, ORDV
     STH    5, FMET
$FORT
          TOV=1*NSUM
$ASSM
     LIS    5,1
     STH    5, FF
     STH    5, I1SIG
     LIS    6,10
     STH    6, NSAM
     CH    5, OPR
     BFC    3, INMET
     LIS    5,5
     STH    5, NSAM
INMET    L    15, R15SAVE
     BFCR    0,15
CKMTFG    ST    15, R15SAVE
     LH 4, OPR   OPR=1
     BFC  3, $P30    NO
     LH   4, FLAG    FLAG =1
     BTC    3, CKM2    YES
```

```
    LIS   4,0   NO
    STH   4,P1
    STH   4,P2
    BFC  0,EXITCKMT
CKM2   LH   5,FORD
    CHI   5,1
    BNE   EXITCKMT
    LIS   5,0
    STH   5,FORD
    LE   0,TM
    FXR   6,0   FIX TIME
    LH   7,OVL   GET ORD VALUE
$FORT
      IF (ICAL.EQ.4) GO TO 30
      CALL SIGLEV(I1SIG,D11)
30    CONTINUE
$ASSM
EXITCKMT   L   15,R15SAVE
    BFCR   0,15
     ALIGN 4
R15SAVE   DS 4
$FORT
C**************START HERE*****************
5      CONTINUE
$ASSM
      L  5,START
    AHI  5,X'B6B'
    ST 5,LAST
   ST  5,PRAM2
   NLIST
$FORT
      IGAS=0
      WRITE(5,810)
      READ(5,720) JC
C
C    JC =0  RL TIME,  JC=1 COPY   ,JC=2  CAL WITH TAPE
C/*
      IF(JC.EQ.0)GO TO 67
   7   PAUSE 10
      EOF=0
$ASSM
   SVC  1,READ
   LH 4,STATUS
   STH 4,EOF
$FORT
      IF (JC.EQ.1) GO TO 40
      CONTINUE
$ASSM
    BAL   15,INITMET
$FORT
```

```
40    CONTINUE
      IF(EOF.EQ.-30587) GO TO 145
      CONTINUE
$ASSM
   BAL   15,PRINTHED
$FORT
      IF (JC.EQ.1) GO TO 9
      IF (OPR.NE.1) GO TO 9
      D11=DLOKI
      ICAL=1
      CALL LOKTEM(ICAL,CRATIO)
      GO TO 9
C   NO SONDES IN THIS PROGRAM
8     STOP
      D11=DSONDE
9     CONTINUE
      TM=TSTAR
      DO 10 I=1,3
      ITM(I)=INT(TM/C(I))
 10   TM=TM-ITM(I)*C(I)
$ASSM
    BAL   15,PRINTHED
$FORT
      WRITE(3,790)(ITM(I),I=1,3)
      FLAG=0
      INDEX=-1
        MINDX=0
      IBFUL=0
      IDSP=0
      EOF=0
11    CONTINUE
$ASSM
   BAL   15,READTP
$FORT
      IF (EOF.EQ.-30587) GO TO 145
      IF (JC.EQ.1) GO TO 45
13    CONTINUE
$ASSM
   BAL   15,CKMTFG
$FORT
45    CONTINUE
      IF (FLAG.EQ.0) GO TO 11
   18 TM=TLOCK
      DO 19 I=1,3
      ITM(I)=INT(TM/C(I))
 19   TM=TM-ITM(I)*C(I)
      WRITE(3,800)(ITM(I),I=1,3)
      IHD=1
      IF(OPR.EQ.1)IHD=2
20    WRITE(3,740) POS(IHD),NSH(IHD),ENH(IHD)
```

```
      IF(JC.EQ.2) GO TO 79
      GO TO 25
22    EOF=0
$ASSM
  SVC  1,READ
  LH   4,STATUS
  STH  4,EOF
$FORT
      IF(EOF.EQ.-30587) GO TO 145
 25   I=0
23    I=I+1
      IF(I.GT.LMETDX) GO TO 22
      WRITE(3,820)RTT(I),RZZ(I),RZF(I),RVY(I),RVX(I),RVS(I),RVD(I)
     1,RVZ(I),RSHEAR(I),RTIMM(I),RSMOT9(I),RSMOT8(I)
      GO TO 23
145   WRITE(5,830)
      READ(5,720) I
      IF(I.EQ.1) GO TO 7
      STOP
 67    CONTINUE
$ASSM
   L   5,NOAFPSW
   EPSR  4,5  PREVENT ARITH FAULT INT
$FORT
      TSTAR=0
      WRITE(5,750)
      READ(5,760)ROCKT,ROUNDN,DY,MN,YR,TIMEL
      WRITE (5,700)
      READ(5,710) OPR,OPQ
      CONTINUE
$ASSM
   BAL   15,PRINTHED
$FORT
      IF (OPR.NE.1) GO TO 69
      ICAL=1
      CALL LOKTEM(ICAL,CRATIO)
      GO TO 69
C   NO SONDES IN THIS PROGRAM
68    STOP
69    CONTINUE
      IHD=1
      IF(OPR.EQ.1)IHD=2
      CONTINUE
$ASSM
   BAL   15,PRINTHED
$FORT
      WRITE(3,740) POS(IHD),NSH(IHD),EWH(IHD)
      WRITE(5,770)
      READ(5,720) IDISP
$ASSM
```

```
    BAL   15,INITMET
   BAL  15,INITIAL    START I/O  BUT NO MAG TAPE
    SVC  1,WRTL52
RDS1   BAL   15,MTPTST
    BAL   15,ENCDDISP
    LIS  13,1
    LHI  14,X'80'
    OCR   13,14
    RHR  13,10
    CHI  10,X'200'
    BTC  3,RDS1
    STH  13,IDISP
    LIS  5,0
    STH  5,FLAG
    STH  5,INDEX  INIT  TAPE
    STH  5,MINDX  BUFFER POINTERS
   BAL  15,READTP
$FORT
      TSTAR = TM
      DO 71 I=1,3
      ITM(I)=INT(TM/C(I))
71    TM=TM-ITM(I)*C(I)
      ENCODE(BUFF,790)(ITM(I),I=1,3)
$ASSM
   SVC 1,WRTL3
    SVC  1,WRTL5
RDS2  BAL  15,MTPTST
    BAL   15,ENCDDISP
    BAL   15,CKMTFG
    LIS  13,1
    LHI  14,X'80'
    OCR  13,14
    RHR  13,10
    CHI  10,X'300'
    BTC  3,RDS2
    STH  13,FLAG
$FORT
76    MINDX=INDEX
$ASSM
   BAL  15,READTP
$FORT
      TLOCK=TM
      DO  77 I=1,3
      ITM(I)=INT(TM/C(I))
77    TM=TM-ITM(I)*C(I)
      ENCODE(BUFF,800)(ITM(I),I=1,3)
$ASSM
   SVC 1,WRTL3
    SVC 1,WRTL53
$FORT
```

```
      GO TO 100
79    CONTINUE
$ASSM
  LHI 5,X'2803'  WRT ASC PROC
  STH 5,WRTL3
  LHI  5,X'2806'
  STH 5,WRTL6
  LCS 5,1
  STH 5,INDEX  INDEX=-1
  LIS 5,0
  STH 5,MINDX  MINDX=0
$FORT
 100  KYY=0
      KA1=1
      IPRNT=1
      AA=0
      IFRST=1
      SHEAR=0.
      IOF=0
230   CONTINUE
      DO 220 I=KA1,121
$ASSM
   BAL   15,CKMTFG
   BAL   15,READTP
$FORT
       IF(IOF.EQ.-30587) GO TO 680
      RAW(1,I)=TM
      IRAW(2,I)=RG
      IRAW(3,I)=AZ
220   IRAW(4,I)=EL
$ASSM
FGWNDCAL EQU *
$FORT
      CALL WNDCAL
      GO TO 230
680   CONTINUE
$ASSM
   SVC   1,BKFL
   SVC   1,BKRC
   SVC   1,READ
$FORT
690   IF (OPR.NE.1) GO TO 691
      IF (ICAL.EQ.4) GO TO 691
      CONTINUE
$ASSM
   BAL   15,PRINTHED
$FORT
      WRITE(3,785)
      I1SIG=-1
      CALL SIGLEV(I1SIG,D11)
```

```
         CONTINUE
$ASSM
     BAL    15,PRINTHED
$FORT
         CALL TEMCOR(NODF,BLALT,BLPR,BLTMP)
         IF (JC.NE.0) GO TO 695
         CONTINUE
$ASSM
     SVC    1,PRAM
*      WRITE LAST RECORD ON MAG TAPE WITH
*      BASE LEVEL VALUES, IF AVAILABLE
$FORT
695      CONTINUE
         CALL SLML
         CONTINUE
         CALL ROCOB
         CONTINUE
691      IF (JC.NE.2) GO TO 1000
         CONTINUE
$ASSM
     SVC    1,FWFL
$FORT
1000     STOP
300      FORMAT('ORD',1X,2F8.3,I7,F7.1)
700      FORMAT('  ROCKET? 1CR,  SONDE? CR')
710      FORMAT (I1,1X,I1)
720      FORMAT(I1)
$ASSM
LAUNST   DC   C'PRESS DTA 2  AT LAUN'
ENDLAU    DC   C'CH'
LOCKST   DC   C'PRESS DTA 3 AT RADAR LOC'
ENDLOC    DC   C'K '
STPST   DC   C'PRESS DTA 8 TO END FLIGH'
ENDSTP   DC   C'T '
$FORT
740        FORMAT('0  TIME     ALTITUDE(MSL)  WIND VELOCITY(M/SEC)    DIR     F
        1L VEL WND SHEAR     TIME          ',A8,  '              ANGLES(DEG)'
        1                                  /'    (SEC)     (KM)     (KFT)    -N+S'
        2,3X,'-E+W    TOTAL   (DEG)     (M/SC)   (MPS/M)   (MIN.SEC)    '
        3,A4,6X,A4,7X,'EL',8X,'AZ')
750      FORMAT( 1H ,32H INPUT RNAME'RDNM'DY'MN'YR'LNTM'    )
760      FORMAT(3A4,1X,I4,1X,I2,1X,I2,1X,I2,1X,I4)
770      FORMAT(1H ,'PRESS CR TO START DISPLAY')
780      FORMAT(1H1,3A4,' NUMBER ',I4,' LAUNCHED ',I2,A4,1X,I2,
        11X,I4,' FROM POKER FLAT '/'STATION LATITUDE',F7.3,' LONGITUDE
        2 ',F9.3, ' ALTITUDE ',F5.0,' METERS SYSTEM AVASS')
730      FORMAT(1H ,'GEOMETRIC ALTITUDES')
735      FORMAT(1H ,'GEOPOTENTIAL ALTITUDES')
785      FORMAT(1H0,4X,'***SIGNIFICANT LEVEL PRINTOUT***',/)
790      FORMAT(' ZULU LAUNCH TIME  ',I2,1H:,I2,1H:,I2)
```

```
800    FORMAT(' RADAR LOCK TIME=  ',I2,1H:,I2,1H:,I2)
810    FORMAT('  REAL TIME CR  PRINT COPY 1CR  CAL WITH MAG TAPE 2CR ')
820    FORMAT(1H ,F7.1,2F8.1,3F7.1,F8.1,F9.0,F8.3,1X,F9.2,2F10.2,2F10.3)
830    FORMAT('  DONE? YES CR,  NO 1CR  ')
$ASSM
READTP   EQU  *
   STM 0,REGS
$FORT
840    DO 860 J=1,10000
$ASSM
   BAL  15,MTPTST
   BAL   15,ENCDDISP
$FORT
 850   IF(INDEX.NE.MINDX) GO TO 870
860    CONTINUE
$ASSM
  BAL  15,INITIAL      RESTART INTERUPTS
$FORT
       GO TO 840
 870   CONTINUE
$ASSM
       LH  13,MINDX
   NLIST
        L  14,RANGE(13)
   CI 14,F'182880'    MAX RADAR RANGE(METERS)
  BFC 2,SF  GREATER?
   L  14,RGL   YES
SF  ST 14,RGL   NO
  SLLS 14,10
  ST 14,RG
       L   14,ELEV(13)
  SLLS 14,11
  ST 14,EL
       L   14,AZMU(13)
  SLLS 14,11
  ST 14,AZ
  NLIST
       LH  14,TIM2(13)     X'0000THHTMM'
       EXHR 14,14          X'THHTMM0000'
       RLL  14,X'4'        X'HTMM0000TH'
       STH  14,ITH
       RLL  14,X'4'        X'TMM0000THH'
       NHI  14,X'F00F'     X'TMM000000H'
       STH  14,IH
       RLL  14,X'4'
       NHI  14,X'F00F'
       STH  14,ITMS
       RLL  14,X'4'
       NHI  14,X'F00F'
       STH  14,IM
```

```
        LH  14,TIM3(13)
        EXHR 14,14
        RLL 14,X'4'
        STH 14,ITS
        RLL 14,X'4'
        NHI 14,X'F00F'
        STH 14,IS
        RLL 14,X'4'
        NHI 14,X'F00F'
        STH 14,IPS
     LR  4,13  SET UP REG FOR TAPE CAL
        AHI 13,H'24'
        STH 13,MINDX
  NLIST
$FORT
        IF(EL.GE.134217728)EL=EL-268435456
        IF(AZ.GE.134217728)AZ=AZ-268435456
        IF(MINDX.EQ.2400) MINDX=0
         TMA  =(ITH*10.+IH)*3600.+(ITMS*10.+IM)*60.+ITS*10.+IS+IPS*.1
        IF (TSTAR.GT.TMA) TMA=TMA+86400.
        TM=TMA-TSTAR
        IF(JC.EQ.0) GO TO 900
        GO TO 4
        IF(TM.LT.(TLOCK+OFS))GO TO 840
 875    IF(MINDX.NE.0) GO TO 890
880     CONTINUE
$ASSM
   SVC 1,READ
   LH 5,STATUS
   STH 5,IOF
   LCS 5,1
   STH 5,INDEX
   LIS 5,0
   STH 5,MINDX
$FORT
890     CONTINUE
$ASSM
   LM 0,REGS
   BFCR 0,15    RETURN
$FORT
900     CONTINUE
$ASSM
   LIS  13,1    SWITCH ADR
   LHI  14,X'80'
   OCR  13,14
   RHR  13,14
   CHI  14,X'800'
   BFC  3,RESET    END OF FLIGHT
   LM  0,REGS
    BFCR  0,15
```

```
     ALIGN  ADC
REGS   DAS   16
CRAD   DC  E'4.79369E-05'
RGL    DS  4
       DS 20   ADDITIONAL CONT STORAGE
RESET   LHI   5,1
   CH   5,IBFUL
   BTC   3,RESET
   LHI 5,X'8D'
  LHI 6,X'C0'   DISARM CARD
  OCR 5,6
  LHI  5,X'8E'
  OCR  5,6  DISARM DISP CARD
   LHI   5,X'8C'
   OCR   5,6   DISARM CLOCK
$FORT
      GO TO 690
$ASSM
   ALIGN 4
READ  DC X'5801'    READ BIN WAIT LU 1(MAG TP)
STATUS  DS 2
START  DC A(ROCKT)
LAST    DC A(LTRAN)+1
    DS 4
    DS 4
   ALIGN  4
WRTL3  DC X'2206'  X'2206' WRITE ASC UNCOND PROC
L3S  DS 2
  DC A(BUFF)
  DC  A(EB)
  DAS 1
  DAS 1
  ALIGN 4
WRTL6  DC  X'2206'
  DS 2
  DC A(BUFF1)
  DC  A(EB1)
  DAS 1
  DAS 1
  ALIGN 4
WRTL5  DC  X'2205'
WRTL5S  DS  2
  DC  A(LOCKST)
  DC A(ENDLOC)
  DAS  1
  DAS  1
WRTL52  DC  X'2205'
WRT52S  DS  2
   DC  A(LAUNST)
   DC  A(ENDLAU)
```

```
       DAS  1
       DAS  1
WRTL53  DC  X'2205'
WRT535  DS  2
       DC  A(STPST)
       DC  A(ENDSTP)
       DAS  1
       DAS  1
NOAFPSW    DC  Y'62F0'   NO ARITH F INT
        ALIGN 4
BKFL  DC  X'8201'
      DS  2
BKRC  DC  X'A001'
      DS  2
FWFL  DC  X'8401'
      DS  2
 LIST
$FORT
         END
```

AD-A053 608 NEW MEXICO STATE UNIV LAS CRUCES F/G 4/2

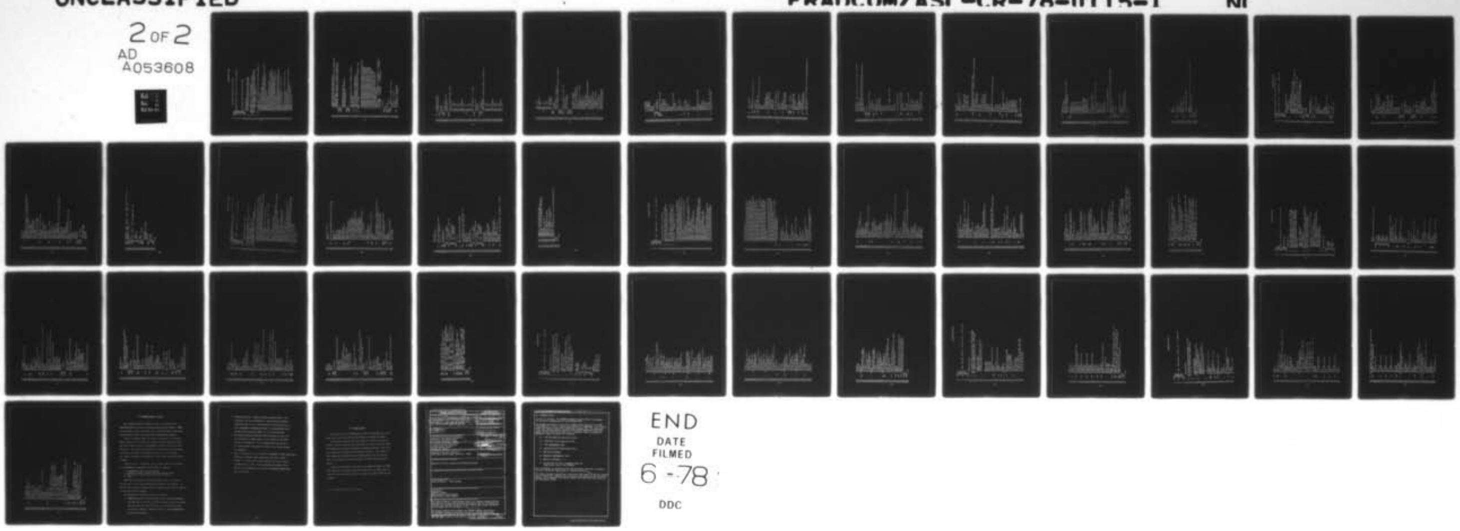
DEVELOPMENT OF A REAL-TIME ROCKETSONDE AND A REAL-TIME RADIOSON--ETC(U)

JAN 78 M D MERRILL, S FRY DAAD07-76-C-0115

UNCLASSIFIED ERADCOM/ASL-CR-78-0115-1 NL

2 OF 2

AD
A053608



END

DATE
FILMED

6-78

DDC

SUBROUTINE WNDCAL

```
C            MET ROCKET AND SONDE WIND CALCULATION
$ASSM
 SCRAT
WNDCOMPC PROG LOKI WIND CAL ROUTINE  03 MAY 77 R01 S WNDCOMP.FOR
  CROSS
  NOSQZ
 NLSTC
  NLIST
  EXTRN  WRTL6,WRTL3,ELOFFS,AZOFFS,MTPTST,ENCDDISP
  NORX3
$FORT
      SUBROUTINE WNDCAL
C     THIS PROGRAM COPIES PREVIOUS DATA CALCULATIONS
C      OR CALCULATES WINDS FOR ROCKETS OR RADIOSONDS
C
C  INDEX CURRENT CELL TO STORE NEXT RAW DATA
C  MINDX CURRENT CELL TO GET DATA FROM(EXT USER UPDATE)
C  METDX CURRENT NUMBER OF STORED CALCULATED MET DATA(USER SET)
C  LMETDX CURRENT NUMBER OF STORED CALCULATED MET DATA IN THIS
C         TAPE  RECORD(SET INTERNALLY)
C
C  NEXT 8 CARDS MUST BE IN ORDER ROCKT,TIMEL,TSTAR,FLAG,RT1,RSHEAR,MET,
C       RANGE,ELERR
      REAL ROCKT(3),TSTAR,TLOCK
       INTEGER*2 TIMEL,MN,DY,YR,OPR,OPQ
      INTEGER*2 FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX
      DIMENSION RT1(10),RZZ(10),RZF(10),RVY(10),RVX(10),RVD(10),RVZ(10),
     1 RSHEAR(10),RTIMM(10),RSMOT9(10),RSMOT8(10),RVS(10)
      INTEGER*2 MET,TIM1,TIM2,TIM3
      REAL RANGE,ELEV,AZMU
      INTEGER*2 ELERR,AZERR,METRAN(1187),LTRAN
      COMMON /BUFDTA/ROCKT,TSTAR,TLOCK,TIMEL,MN,DY,YR,OPR,OPQ
     1,FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX,RTT,RZZ,RZF,RVY,
     2 RVX,RVD,RVZ,RSHEAR,RTIMM,RSMOT9,RSMOT8,RVS,MET,TIM1,TIM2
     3,TIM3,RANGE,ELEV,AZMU,ELERR,AZERR,METRAN,LTRAN
      COMMON /BLOCK2/VSA,VDA,CORTPK,TTPA,ZMA,TPCOR,PRSS,DENS
     1,VZA,SPSND,VYUA,VXUA
      DIMENSION TTPA(60),ZMA(60),VSA(60),VDA(60),VZA(60)
      DIMENSION TPCOR(60),CORTPK(60),PRSS(60),DENS(60),SPSND(60)
      DIMENSION VYUA(60),VXUA(60)
      DIMENSION D(3),CIF(3),F(3),G(3),H(3),D1(3),VU(3)
      INTEGER*2  KA4,L,I,IRS,I72
      REAL G1,G2
      INTEGER*4 IRAW(4,121),ISMOT(4,5),IWGT(59),ID(3),ID1(3)
      DIMENSION   WGT(59),WW(10)
      DIMENSION RAW(4,121),SMOT(4,5)
      DIMENSION SPX(3),SPY(3),SZZ(3),STT(3),SAZ(3),SEL(3),AV(10)
      DIMENSION  BUFF(33),ACC(3,10),BUFF1(33)
      INTEGER*2 I1,J,J1,J2,K,KA,KA1,KA2
      INTEGER*2 KYY,I1,LL,NZ
```

```
      INTEGER*2 AA, IFRST, EB, EB1, ICODE, IDSP, JC
     COMMON /BLOCK1/ AA, IFRST, IPRNT, IRAN, KA1, KYY, BUFF, EB, BUFF1, EB1,
    1 IBFULL, JC, IDSP
     EQUIVALENCE (IRAN(1,1),RAN(1,1)),(SMOT(1,1),ISMOT(1,1))
     EQUIVALENCE (WGT(1), IWGT(1)), (D(1), ID(1)), (D1(1), ID1(1))
$ASSM
   NLIST
$FORT
     EQUIVALENCE(VXU,PX),(VYU,PY)
     DATA CIF/8270.,342313216.,342313216./
     DATA ELOFS,AZOFS/0.,0./
$ASSM
  NLIST
$FORT
     DATA G1,G2,CRDTDG/1.40167E7,1.2742458E7,57.2957755/
     DATA SALT/412./
C WS  DATA SALT/1362./
     DATA C1,GG,RE/.0174533,9.822888,6370233./
C WS   DATA C1,GG,RE/6348612.,9.795168,6347250./
     DATA DHTL,DHTS,CMTF,CMPSTK/100.,4169.1,3.2808399,1.9428/
C   MERRILL  FILTER
C
     DATA IWGT/706358135,705508911,702966643,698747490,692878216
    1,685395947,676347825,665790574,653789982,640420300
    2,625763564,609908853,592951489,574992183,556136144
    3,536492153,516171618,495287616,473953933,452284118
    4,430390549,408383532,386370439,364454888,342735983
    5,321307613,300257829,279668280,259613751,240161766
    6,221372292,203297532,185981799,169461494,153765164
    7,138913655,124920342,111791449,99526436,88118457
    8,77554884,67817886,58885050,50730046,43323322
    9,36632815,30624675,25263995,20515527,16344380
    A,12716701,9600310,6965298,4784571,3034342
    B,1694547,749209,186711,0/
     DATA  WW/        0.34506476E-01,  0.64746658E-01,  0.89803061E-01,
    1 0.10896906,     0.12177899,      0.12803132,      0.12780346,
    2 0.12145806,     0.10963999,      0.93262928E-01/
C
     GO TO 325
$ASSM
   ALIGN  4
R15S  DS  4
ARCT  ST 15,R15S
$FORT
  306 IF ((ABS(VY)).GT..0001) GO TO 310
     VD=90.
     IF (VX.GT.0.) VD=270
     GO TO 320
310   VD=(ATAN2(VX,VY))*CRDTDG+180
     IF (VD.GT.360.) VD=VD-360.
```

```
320     VS=SQRT(VX**2+VY**2)
        VT=SQRT(VS**2+VZ**2)
$ASSM
    L 15,R15S
   BFCR 0,15
$FORT
 325  DO 350 J=1,5
        SMOT(1,J)=RAN(1,J+58)
        DO 350 K=2,4
$ASSM
*       ISMOT(K,J)=IWGT(1)*IRAN(K,J+58)
        LDA  11,IWGT
        LH    7,J
        AHI   7,58
        LIS   5,4
        MAR   6,5
        AH    7,K
        SIS   7,5
        SLLS  7,2
        LDA   3,IRAN(7)
        MAR   10,3
        LH    7,J
    LIS 5,4
        MAR   6,5
        AH    7,K
        SIS   7,5
        SLLS  7,2
        STA   10,ISMOT(7)    ST 32 MSBS
$FORT
        DO 350 L= 2,59
        J1=J+L+57
        J2=J-L+59
$ASSM
   BAL  15,MTPTST
   BAL    15,ENCDDISP
*       ISMOT(K,J)=ISMOT(K,J)+IWGT(L)*(IRAN(K,J1)+IRAN(K,J2))
        LH    7,J1
        LIS   5,4
        MAR   6,5
        AH    7,K
        SIS   7,5
        SLLS  7,2
        LDA  11,IRAW(7)
        LH    7,J2
        MAR   6,5
        AH    7,K
        SIS   7,5
        SLLS  7,2
        AA   11,IRAW(7)
        LH    7,L
```

```
        SIS   7,1
        SLLS  7,2
        LDA   3,IWGT(7)
        MAR  10,3
        LH    7,J
     LIS 5,4
        MAR   6,5
        AH    7,K
        SIS   7,5
        SLLS  7,2
        AA   10,ISMOT(7)
        STA  10,ISMOT(7)
$FORT
350     CONTINUE
        DO 360 I=2,4
        ID(I-1)=ISMOT(I,4)-ISMOT(I,2)
        ID1(I-1)=ISMOT(I,5)-ISMOT(I,3)+ISMOT(I,1)-ISMOT(I,3)
$ASSM
 NLIST
$FORT
360     SMOT(I,3)=ISMOT(I,3)/CIF(I-1)
        DO 370 I=1,3
        D(I)=ID(I)/CIF(I)
370     D1(I)=ID1(I)/CIF(I)
$ASSM
 NLIST
$FORT
        ELT=SMOT(4,3)*CRDTDG
        AZT=SMOT(3,3)*CRDTDG
        IF(AZT.LT.0.)AZT=360.+AZT
        IF (AZT.GT.360.) AZT=AZT-360.
        TT=SMOT(1,3)
C       USE PARTIAL DER TO CAL VELOCIT
        SE=SIN(SMOT(4,3))
        A=SE*SMOT(2,3)
        CE=COS(SMOT(4,3))
        B=CE*SMOT(2,3)
        ZZ=(B*B)/G2 +A+SALT
C          CORRECT HDO FOR EARTHS CURVATURE
        HDO= B-(B*ZZ)/G1
        SA=SIN(SMOT(3,3))
        CA=COS(SMOT(3,3))
C        COMPUTE  EW POSITION
        H(2)=-HDO*SA
C          COMPUTE NS POSITION
        H(1)=HDO*CA
        F(3)=SE+2.*B*CE/G2
        G(3)=B-2.*B*A/G2
        DDDR=CE-(B*F(3)+ZZ*CE)/G1
        DDDE=-A-(B*G(3)-ZZ*A)/G1
```

```
      F(1)=DDDR*SA
      F(2)=DDDR*CA
      G(1)=DDDE*SA
      G(2)=DDDE*CA
      H(3)=0
      PY=H(1)
      PX=-H(2)
$ASSM
   BAL  15,MTPTS1
   BAL   15,ENCDDISP
$FORT
380   IF(IPRNT.NE.1)GO TO 390
80    STT(1)=0
      SPX(1)=-683.84
      SPY(1)=1014.78
      SZZ(1)=197.73
      SEL(1)=0
      SAZ(1)=0
      IRS=2
      DL=60.
      TST=0.
      IF(OPR.NE.1) GO TO 385
      HTS=125000
      LL=0
      FLV=0.
      DO 382 I=1,60
38.   TTPA(I)=-1.
      IRS=1
      VT=0.
 385  DO 387 I=1,3
      DO 387 K=1,10
      AV(K)=0.
387   ACC(I,K)=0.
      TTP=TT
      TL=TT
      VX=0.
      VY=0.
      VZ=0.
      VS=0.
C     HDO NEEDED FOR LOKI FLIGHT
      VD=HDO
      SHEAR=0.
      RSHEAR(9)=0.
      RSHEAR(10)=0.
      SAZ(2)=AZT
      SEL(2)=ELT
      IZ=ZZ/100.
      GO TO 620
C     VU(I) UNCOR VEL 1,X;2,Y;3,Z
 390  IF(OPR.NE.1) GO TO 520
```

```
C
C     START LOKI PROG
C
      DO 400 I=1,3
400   VU(I)=(F(I)*D(1)+G(I)*D(3)+H(I)*D(2))/.2
C
 410  DO 430 K=1,3
      DO 420 J=1,9
420   ACC(K,J)=ACC(K,J+1)
430   ACC(K,10)=(F(K)*D1(1)+G(K)*D1(3)+H(K)*D1(2))/.2
      LL=LL+1
      XAC=0.
      YAC=0.
      ZAC=0.
      DO 440 J=1,10
      XAC=XAC+WW(J)*ACC(1,J)
      YAC=YAC+WW(J)*ACC(2,J)
440   ZAC=ZAC+WW(J)*ACC(3,J)
      IF (LL.GE.10) GO TO 450
      XAC=ACC(1,10)
      YAC=ACC(2,10)
      ZAC=ACC(3,10)
C
C   CHECK FOR WIND SENSITIVITY
C
      IF(VU(3).GE.FLV)GO TO 510
      FLV=VU(3)
      LL=0
      GO TO 510
450   DEN=ZAC+GG*.2
      VXC=VU(1)-VU(3)*XAC/DEN
      VYC=VU(2)-VU(3)*YAC/DEN
      VX=VXC
      VY=VYC
$ASSM
   BAL  15,ARCT   GET ANGLE
$FORT
      IF(AA.NE.0) GO TO 470
 460  IF(HTS.LT.22)GO TO 470
      HTS=HTS-1000
      AA=1
      GO TO 460
 470  IF(ZZ.GT.HTS)GO TO 474
      HTS=HTS-1000
      GO TO 560
 474  IF(IRS.EQ.1)GO TO 670
C      MAKE Q A CHECKS ON DATA
C
C      CHECK FOR CHANGE IN TOTAL VELOCITY, SHIFT IN DIRECTION,SPEED CHAN  A 2
 480  VT2=1.3*VT1
```

```
          VT3=0.7*VT1
          IF((VS.GT.10.).AND.((VT.GT.VT2).OR.(VT.LT.VT3))) GO TO 510
          Z=ZZ/1000.
          ZF=Z*3.2808399
          SHEAR=SQRT((VX-VX1)**2+(VY-VY1)**2)/(ZZ1-ZZ)
          ENCODE(BUFF1,820)TT,Z,ZF,VY,VX,VS,VD,VU(3),SHEAR,TIMM,VU(2),
         1VU(1)
$ASSM
 SVC  1,WRTL6
$FORT
  490   IF(IFRST.EQ.0) GO TO 505
        DO 500 K=1,5
        AV(K)=AV(K+5)
 500    AV(K+5)=0.
        IFRST=0
  505   AV(6)=AV(6)+VX
        AV(7)=AV(7)+VY
        AV(8)=AV(8)+VU(3)
        AV(10)=AV(10)+1.
  510   ZZ1=ZZ
        VX1=VX
        VY1=VY
        VD1=VD
        VS1=VS
        VT1=VT
        IF (ZZ.LT.56000.) GO TO 670
        KA=116
        KA2=5
        KA1=117
        GO TO 675
C
C       START RADIOSONDE PROG
C
  520   IF(AA.NE.0) GO TO 550
 530    IF(TST.GT.TL)GO TO 540
        TST=TST+DL
        AA=1
        GO TO 530
 540    TL=TST-DL
 550    IF(TT.LT.TL) GO TO 670
        TL=TL+DL
 560    SPX(IRS)=-H(2)
        SPY(IRS)=H(1)
        STT(IRS)=TT
        SZZ(IRS)=ZZ
        SEL(IRS)=ELT
        SAZ(IRS)=AZT
   570  IRS=IRS+1
        IF(OPR.EQ.1) GO TO 580
        IF(IRS.LT.4) GO TO 670
```

```
      GO TO 590
 580  IFRST=1
      IF(IRS.LT.4) GO TO 480
      DN=AV(5)+AV(10)
      VXT=VXB
      VYT=VYB
      IF(DN.EQ.0.)GO TO 490
      VX=(AV(1)+AV(6))/DN
      VY=(AV(2)+AV(7))/DN
      VZ=(AV(3)+AV(8))/DN
      VXB=VX
      VYB=VY
      SHEAR=(SQRT((VXT-VXB)**2+(VYT-VYB)**2))/1000
C SMOT(8,3)UNCORRECTED VX,SMOT(9,3)UNCORR VY,2F UNCOR VZ,T1MM UNC SHEAR
      DN=STT(3)-STT(1)
      VXU=(SPX(3)-SPX(1))/DN
      VYU=(SPY(3)-SPY(1))/DN
      GO TO 610
 590  CONTINUE
      DN=(STT(3)-STT(1))
      VX=(SPX(3)-SPX(1))/DN
      VY=(SPY(3)-SPY(1))/DN
      VZ=((SZZ(3)-SZZ(1))/DN)
      ZZ1=SZZ(1)
      ZZ2=SZZ(3)
      PX=SPX(2)
  600 PY=SPY(2)
      SHEAR=SQRT((VX-VX1)**2+(VY-VY1)**2)/(ZZ2-ZZ1)
      VX1=VX
      VY1=VY
      TTP=STT(2)
      ZE=SZZ(2)
      GO TO 715
  610 TTP=STT(2)
      ZE=HTS
 713  IF(ZE.GE.SZZ(2)) GO TO 715
      ZE=ZE+1000
      GO TO 713
  715 IZ=INT(ZE/100.+.5)
      DO 620 K=1,2
      SPX(K)=SPX(K+1)
      SEL(K)=SEL(K+1)
      SAZ(K)=SAZ(K+1)
      SPY(K)=SPY(K+1)
      STT(K)=STT(K+1)
  620 SZZ(K)=SZZ(K+1)
      IRS=3
$ASSM
   BAL 15,ARCT   GET ANGLE
$FORT
```

```
  630  ZM=IZ/10
        ZF=ZM*3.2808399
       TIMM=(TTP+.5)/60
       TIMM=AINT(TIMM)+AMOD(TIMM,1.0)*.6-.008333
C       PREPARE TO PRINT SAMPLE
C
640    CONTINUE
       METDX=METDX+1
       RTT(METDX)=TTP
       RZZ(METDX)=ZM
       RZF(METDX)= ZF
       RVY(METDX)= VY
       RVX(METDX)= VX
       RVS(METDX)= VS
       RVD(METDX)= VD
       RVZ(METDX)=VZ
       RSHEAR(METDX)=SHEAR
       RTIMM(METDX)=TIMM
       RSMO18(METDX)=VYU
       RSMOT9(METDX)=VXU
       KYY=1
       NZ=0
       IF (IPRNT.EQ.1) GO TO 645
       IF(OPR.EQ.0)GO TO 645
       IZ2=ZM-19
       IF((IZ2.GT.60).OR.(IZ2.LT.1))GO TO 645
       TTPA(IZ2)=TTP
       ZMA(IZ2)=ZM
       VSA(IZ2)=VS
       VDA(IZ2)=VD
       VZA(IZ2)=VZ
       VYUA(IZ2)=VYU
       VXUA(IZ2)=VXU
C
C       STORE DATA FOR SUBSEQUENT USE
C
645    ENCODE(BUFF,820)TTP,ZM,ZF,VY,VX,VS,VD,VZ,SHEAR,TIMM,VYU
      1,VXU,SEL(2),SHZ(2)
650    CONTINUE
$ASSM
    SVC   1,WRTL3
$FORT
C
 655   IF(IPRNT.NE.1) GO TO 660
       IPRNT=0
       GO TO 390
 660    IF(OPR.EQ.1) GO TO 480
C       MOVE DATA IN RAW DATA ARRAY
C
670    KA4=20
```

```
      KA=111
      KA1=112
      KA2=10
C
675   KA4=KA4-1
      DO 680 I=1,KA
      I1=I+KA2
      RAN(1,I)=RAN(1,I1)
      DO 680 J=2,4
680   IRAN(J,I)=IRAN(J,I1)
C
C      READ AND STORE 10 NEW SAMPLES
C
      RETURN
820   FORMAT(1H ,F7.1,2F8.1,3F7.1,F8.1,F9.0,F8.3,1X,F9.2,2F10.2,2F10.3)
830   FORMAT(1H ,6F8.3,9F8.1)
$ASSM
  LIST
$FORT
      END
```

SUBROUTINE SIGLEV

```
C  PROGRAM TO CAL UNCOR LOKI TEMP  FILE MTTR:SIGLEV.FOR
$ASSM
  SCRAT
SIGLEV  PROG  26 APR 77  R02    FIND UNCOR TEMP   S SIGLEV.FOR
  NOSQ2
  NLIST
  NLSTC
  NORX3
$FORT
      SUBROUTINE SIGLEV(I1SIG,D55)
C     I1SIG=1 INITIALIZE
C          =0 NORMAL
C          = -1 LAST VALUE
      DIMENSION CKAA(60),REFTM(60)
      INTEGER*2 VALUE,TIME,VLAST,TLAST,VINITL,TINITL
      INTEGER*2 IK,NORM,REFT,ISTART,ITM(2),I,OFFSET
      INTEGER*2 REFA(75),TMREFA(75),LKTIME(60),JREF,JTMP,JTMPL,JRFC
      COMMON /LKTMTP/LKTMP,LKTIME
      INTEGER*2 LRTM,LKTMP(60),JR,ICAL,VL,VLT,VU,VUT,I1SIG
      REAL M00,M11,M22,M33,M44,PR(33)
      DATA  IK,NORM,REFT,OFFSET/1024,8192,8000,248/
C WS  DATA  IK,NORM,REFT,OFFSET/1024,8192,9000,248/
      DATA D11/180./
10    CONTINUE
$ASSM
   STH  6,TIME
  STH  7,VALUE
  NLIST
$FORT
      VALUE=VALUE+OFFSET
      ENCODE(PR,1200)TIME,VALUE
$ASSM
   SVC   1,PRTORD
$FORT
      IF(I1SIG.NE.1)GO TO 100
      I1SIG=0
      DO 20 IK=1,60
20    LKTIME(IK)= -1
      LRTM=0
      JTMP=0
      JREF=0
      M22=10000
      M11=-10000
  50  IF(VALUE.GT.REFT) GO TO 350
      TINITL=TIME
      VINITL=VALUE
$ASSM
   BAL  15,JTMPINC
$FORT
      LKTMP(JTMP)=VALUE
```

```
      LKTIME(JTMP)=TIME
      JRFC=2
      JTMPL=1
      GO TO 360
100   IF(IISIG.EQ.-1) GO TO 500
      IF(JTMP.EQ.0)GO TO 50
      IF(VALUE.GT.REFT)GO TO 350
      VLAST=VALUE
      TLAST=TIME
 110  D22=TIME-TINITL
      D33=VALUE-VINITL
      M00=D33/D22
      IF(M00.GT.M22)GO TO 140
      IF(M00.LT.M11)GO TO 150
      M44=(D33+D11)/D22
      M33=(D33-D11)/D22
      IF(M22.LE.M44)GO TO 410
      M22=M44
      VU=VALUE
      VUT=TIME
      GO TO 410
140   CONTINUE
$ASSM
   BAL   15,JTMPINC
$FORT
      LKTMP(JTMP)=VU
      LKTIME(JTMP)=VUT
      TINITL=VUT
      VINITL=VU
      GO TO 155
150   CONTINUE
$ASSM
   BAL  15,JTMPINC
$FORT
      LKTMP(JTMP)=VL
      LKTIME(JTMP)=VLT
      TINITL=VLT
      VINITL=VL
  155 D22=TIME-TINITL
      D33=VALUE-VINITL
      M22=(D33+D11)/D22
      M11=(D33-D11)/D22
      VL=VALUE
      VU=VALUE
      VLT=TIME
      VUT=TIME
160   IF(JREF.LT.2) GO TO 360
      IF(JTMPL.GT.JTMP) GO TO 360
170   IF(LKTIME(JTMPL).LT.TMREFA(JRFC))GO TO 175
      IF(JRFC.EQ.JREF)GO TO 360
```

```
      JRFC=JRFC+1
      GO TO 170
  175 JR=JRFC-1
      SLOPE=(REFA(JRFC)*1.-REFA(JR))/(TMREFA(JRFC)-TMREFA(JR))
      REFER=REFA(JR)+(LKTIME(JIMPL)-TMREFA(JR))*SLOPE
      CRATIO=LKTMP(JIMPL)/REFER
      CRAA(JTMPL)=CRATIO
      REFTM(JTMPL)=REFER
      CALL LOKTEM(ICAL,CRATIO)
      LKTMP(JTMPL)=CRATIO*10.
      JTMPL=JTMPL+1
      IF(I1SIG.NE.-1) GO TO 160
      IF(JTMPL.GT.JTMP)GO TO 360
      GO TO 175
350   IF((LRTM+20).GT.TIME)GO TO 360
      JREF=JREF+1
      IF (JREF.GE.76) JREF=75
      REFA(JREF)=VALUE
      TMREFA(JREF)=TIME
      LRTM=TIME
      GO TO 160
360   IF(I1SIG.NE.-1) RETURN
      JIMPL=JTMPL-1
      DO 365 I=1,JIMPL
      TM=LKTIME(I)
      CRATIO=LKTMP(I)/10.
      ITM(1)=TM/60.
      ITM(2)=TM-ITM(1)*60.
      WRITE(3,1000)TM,ITM(1),ITM(2),CRATIO,CRAA(I),REFTM(I)
365   CONTINUE
      DO 370 I=1,JREF
      TIME=TMREFA(I)
      VALUE=REFA(I)
      ITM(1)=TIME/60.
      ITM(2)=TIME-ITM(1)*60.
 370  WRITE(3,1100)TIME,ITM(1),ITM(2),VALUE
      RETURN
  410 IF(M11.GT.M33)GO TO 360
      M11=M33
      VL=VALUE
      VLT=TIME
      GO TO 360
500   CONTINUE
$ASSM
   BAL    15,JIMPINC
$FORT
      LKTMP(JTMP)=VLAST
      LKTIME(JIMP)=TLAST
      JRFC=JREF
      GO TO 175
```

```
1000  FORMAT(1H ,'TIME= ',I6,I4,':',I2,'  TEMP= ',F8.1,F12.4,F12.1)
1100  FORMAT(1H ,'TIME= ',I6,I4,':',I2,' REF= ',I6)
1200  FORMAT(1H ,I6,5X,I6)
$ASSM
   ALIGN  4
PRTORD    DC   X'2208'    WRITE ASC UNCOND PROC LU 8
   DS  2
   DC  A(PR)
   DC  A(PR)+131
   DAS  1
   DAS   1
SVE     DAS    1
JTMPINC    ST    15,SVE
$FORT
      JTMP=JTMP+1
      IF(JTMP.GE.61) JTMP=60
      CONTINUE
$ASSM
   L  15,SVE
   BFCR  0,15
  LIST
$FORT
      END
```

SUBROUTINE LOKTEM

```
C  PROGRAM TO CALCULATE UNCORR TEMP FOR LOKI MTTR:LOKI.TEM
C  LINKABLE OBJECT FILE MTTR:LOKTEMP.OBL
$ASSM
  SCRAT
LOKTEMP   PROG     28 APR 77  R01   S MTTR:LOKTEMP.FOR
 NOSOZ
 NORX3
  NLIST
  NLSTC
$FORT
      SUBROUTINE LOKTEM(ICAL,CRATIO)
C
C FOR READING OF CALIBRATION TAPE CALL WITH ICAL=1)
C FOR NORMAL TEMP PROCESSING  ICAL=0
C JC=0  REAL TIME CALL
C JC=2  CALCULATION WITH TAPE CALL
C CALLING PROGRAM SENDS CRATIO(TORD/REFORD)
C SUBR RETURNS UNCORRECTED LOKI TEMPERATURE CRATIO
C
      COMMON /BUFDTA/ROCKT,TSTAR,TLOCK,TIMEL,MN,DY,YR,OHR,OHO
     1,FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX,RTT,RZZ,RZF,RVY
     2,RVX,RVD,RVZ,RSHEAR,RTIMM,RSMOT9,RSMOT8,RVS,MET,TIM1,TIM2
     3,TIM3,RANGE,ELEV,AZMU,ELERR,AZERR,METRAN,LTRAN
      COMMON /BLOCK1/AA,IFRST,IPRNT,IRAN,KA1,KYY,BUFF,EB,BUFF1,EB1
     1,IBFULL,JC,IDSP
      DIMENSION CALDAT(6),RTT(10),RZZ(10),DAT(21)
      REAL A,B,C,D,E,F
      INTEGER*2 RATIOK(21),LOGIR(5),LOGR(21),I,ICRATO
      INTEGER*2 CLOGTR,ICAL,J,M,JC
      REAL ROCKT(2)
      REAL TSTAR,TLOCK
      INTEGER*2 TIMEL,MN,DY,YR,OHR,OHO
      INTEGER*2 FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX
      DIMENSION RZF(10),RVY(10),RVX(10),RVD(10),RVZ(10),RSHEAR(10)
     1,RTIMM(10),RSMOT9(10),RSMOT8(10),RVS(10)
      INTEGER*2 MET,TIM1,TIM2,TIM3
      REAL RANGE,ELEV,AZMU
      INTEGER*2 ELERR,AZERR,METRAN(1187),LTRAN
      INTEGER*2 AA,IFRST,KA1,KYY,EB,EB1,IDSP
      INTEGER*4 IRAN(4,121)
      DIMENSION BUFF(33),BUFF1(33)
      EQUIVALENCE (CALDAT(1),A),(CALDAT(2),B),(CALDAT(3),C)
     1,(CALDAT(4),D),(CALDAT(5),E),(CALDAT(6),F)
      DATA LOGR/0,1609,2203,2996,3401,3689,3912,4094,4248,4382,
     1 4500,4605,5011,5298,5704,5991,6397,6685,6908,7313,7601/
      IF(ICAL.EQ.-1) GO TO 45
   10 IF(ICAL.NE.1) GO TO 60
      IF (JC.EQ.2) GO TO 140
14    WRITE (5,650)
      READ (5,800)I
```

```
      IF (I.NE.1) GO TO 15
      ICAL=4
C
C   FLAG TO CALLING PROGRAM THAT CAL DATA IS NOT THERE
C
      RZZ(1)=-1.0
      RETURN
15    RTT(9)=-1.0
      RTT(10)=-1.0
      WRITE(5,700)
      READ(5,800)I
      DO 20 I=1,21
      READ(2,900) RAT
20    RATIOK(I)=1000.*RAT
      DO 40 I=1,5
      READ(2,900) RAT
40    LOGTR(I)=1000.*ALOG(RAT)
  45  S1=15./(LOGTR(1)-LOGTR(2))
      S2=-25./(LOGTR(2)-LOGTR(3))
      C=(S1+S2)/(LOGTR(3)-LOGTR(1))
      B=-S1-C*(LOGTR(1)+LOGTR(2))
      A=-65.-B*LOGTR(1)-C*LOGTR(1)*LOGTR(1)
      S3=25./(LOGTR(3)-LOGTR(4))
      S4=-35./(LOGTR(4)-LOGTR(5))
      F=(S3+S4)/(LOGTR(5)-LOGTR(3))
      E=-S3-F*(LOGTR(3)+LOGTR(4))
      D=-25.-E*LOGTR(3)-F*LOGTR(3)*LOGTR(3)
      IF(ICAL.EQ.-1)RETURN
      CLTST=.3
      DO 400  J=1,3
      CRATIO=CLTST
  60  ICRATO=CRATIO*1000.
      DO 200 I=1,21
      IF(ICRATO.GE.RATIOK(I)) GO TO 250
200   CONTINUE
      ICAL=-2
      RETURN
250   FRAC=(ICRATO-RATIOK(I)*1.)/(RATIOK(I-1)-RATIOK(I))
      CLOGTR=LOGR(I)+FRAC*(LOGR(I-1)-LOGR(I))
      IF(CLOGTR.LT.LOGTR(3)) GO TO 300
      CRATIO=A+B*CLOGTR+C*CLOGTR*CLOGTR
      GO TO 350
300   CRATIO=D+E*CLOGTR+F*CLOGTR*CLOGTR
 350  IF(ICAL.EQ.1)GO TO 375
 360  ICAL=0
      RETURN
375   CONTINUE
      WRITE(5,900) CLTST,CRATIO
400   CLTST=CLTST+.2
      WRITE(5,1000)
```

```
      READ(5,800)I
      IF (I.EQ.1) STOP
C
C     STORE CALIBRATION DATA IN BUFFER
C     A,B,C,D,E,F = CALDAT(M)
C
100   DO 110 M=1,6
110   RTT(M)=CALDAT(M)
$ASSM
   LHI   6,H'24'    COUNTER
   LIS   7,0    ADR INDEX
MDTA   LH   3,RATIOK(7)    START ADR
   STH   3,RZZ(7)
   AIS   7,2    INC ADR INDEX
   SIS   6,1    DEC COUNTER
   BNZ   MDTA    MORE DATA
$FORT
      GO TO 175
C
C     READ CALIBRATION DATA FROM BUFFER
C
140   IF ((RZZ(1).EQ.-1.0).OR.((RTT(9).NE.-1.0)
     1.AND.(RTT(10).NE.-1.0))) GO TO 142
      GO TO 150
142   WRITE(5,675)
      GO TO 14
150   DO 160 M=1,6
160   CALDAT(M)=RTT(M)
$ASSM
   LHI   6,H'24'    COUNTER
   LIS   7,0    ADR INDEX
MDT2   LH   3,RZZ(7)    START ADR
   STH   3,RATIOK(7)
   AIS   7,2    INC ADR INDEX
   SIS   6,1    DEC COUNTER
   BNZ   MDT2    MORE DATA
$FORT
175   WRITE(3,750)
      WRITE(3,850)
      DO 180 I=1,21
      DAT(I)=RATIOK(I)/1000.0
180   WRITE(3,855) DAT(I)
      WRITE(3,860)
      DO 185 I=1,5
185   DAT(I)=EXP(LOGTR(I)/1000.0)
      WRITE (3,865) (DAT(I),I=1,5)
      WRITE (3,870)
      WRITE (3,875) A,B,C,D,E,F
      GO TO 360
650   FORMAT(1H ,'CAL TAPE AVAILABLE?  YES PRESS CR ,   NO 1CR')
```

```
675   FORMAT(1H ,'CAL DATA NOT ON MAG TAPE')
700   FORMAT(' PUT LOKI TEMP TAPE IN READER PRESS CR')
750   FORMAT(1H0,/,5X,'***LOKI CALIBRATION TAPE DATA***')
800   FORMAT( I1)
850   FORMAT(1H0,4X,'*RATIOS*')
855   FORMAT(1H ,F10.4)
860   FORMAT(1H0,'*CALIBRATION VALUES*')
865   FORMAT(1H ,5F10.4)
870   FORMAT(1H0,'*QUADRATIC COEFFICIENTS*')
875   FORMAT(1H ,3F10.4,/,1H ,3F10.4)
900   FORMAT(1H ,2F10.4)
1000  FORMAT(1H ,'CAL RATIOS WITHIN .3 DEG? PRESS CR,NO PRESS 1CR')
$ASSM
  LIST
$FORT
      END
```

SUBROUTINE TEMCOR

```
$ASSM
  SCRAT
  NOSQZ
  NORX3
LKTMPCOR   PROG   CORRECTS TEMPS   26 APR 77  R02   S-LKTMPCRT.FOR
  NLSTC
  CROSS
  NLIST
$FORT
      SUBROUTINE TEMCOR(NODF,BLALT,BLPR,BLTMP)
C
C     THIS PROGRAM USES DATA CALCULATED AT EACH EVEN
C     VALUE OF ALTITUDE TO OBTAIN A CORRECTED TEMPERATURE.
C     NODF IS A FLAG TO SIGNAL WHETHER FLIGHT IS AT
C     NIGHT OR DAY.
C     NODF =0  DAY FLIGHT;   =1  NIGHT FLIGHT.
C     THIS ROUTINE ALSO USES DATA FROM RADIOSONDE FOR USE
C     AS BASE LEVEL VALUES IN CALCULATING PRESSURES.
C     THE FLAG THAT INDICATES THAT BASE LEVEL VALUES ARE
C     AVAILABLE IS:     RSHEAR(9),(10) = -1.0
C
      COMMON /BUFDTA/ROCKT,TSTAR,TLOCK,TIMEL,MN,DY,YR,OPR,OPQ
     1,FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX,RTT,RZZ,RZF,RVY
     2,RVX,RVD,RVZ,RSHEAR,RTIMM,RSMOT9,RSMOT8,RVS,MET,TIM1,TIM2
     3,TIM3,RANGE,ELEV,AZMU,ELERR,AZERR,METRAW,LTRAN
      COMMON /BLOCK1/AA,IFRST,IPRNT,IRAW,KA1,KYY,BUFF,EB,BUFF1,EB1
     1,IBFULL,JC,IDSP
      COMMON /BLOCK2/VSA,VDA,CORTPK,TTPA,ZMA,TPCOR,PRSS,DENS
     1,VZA,SPSND,VYUA,VXUA
      COMMON /LKTMP/LKTMP,LKTIME
      DIMENSION TTPA(60),ZMA(60),VSA(60),VDA(60),VZA(60),TPCOR(60)
     1,CORTPK(60),GA(60),PRSS(60),DENS(60),SPSND(60)
      DIMENSION VYUA(60),VXUA(60)
      INTEGER*2 LKTMP(60),LKTIME(60)
      INTEGER*2 K1(51),K2(51),K3(51),K4(51,2)
      INTEGER*2 J,K,IZ,NODF,FL4,KZ,FIZ,LIZ,JZ,IB
      REAL T1,T2,T3,TP1,TP2
      INTEGER*2 TIMEL,MN,DY,YR,OPR,OPQ
      INTEGER*2 FLAG,INDEX,METDX,MINDX,ROUNDN,LMETDX
      DIMENSION RTT(10),RZZ(10),RZF(10),RVY(10),RVX(10),RVD(10)
     1,RVZ(10),RSHEAR(10),RTIMM(10),RSMOT9(10),RSMOT8(10),RVS(10)
      INTEGER*2 MET,TIM1,TIM2,TIM3
      INTEGER*2 ELERR,AZERR,METRAW(1187),LTRAN
      INTEGER*2 AA,IFRST,KA1,KYY,EB,EB1,IDSP,JC
      INTEGER*4 IRAW(4,121)
      DIMENSION BUFF(33),BUFF1(33),ROCKT(3)
      DATA K1/455,454,453,452,452,452,451,450,450,449,449
     1,448,447,447,446,446,445,445,444,444,443,443
     2,442,442,441,441,440,439,439,439,439,440,440,441
     3,442,443,444,446,448,450,452,455,457,459,461,462
```

```
     4, 463, 464, 464, 465, 465/
      DATA K2/535, 550, 565, 580, 597, 616, 635, 656, 677, 700, 724
     1, 749, 776, 804, 837, 868, 903, 940, 978, 1019, 1065, 1111
     2, 1160, 1212, 1271, 1330, 1393, 1469, 1545, 1627, 1728, 1828
     3, 1935, 2053, 2184, 2342, 2510, 2720, 2950, 3230, 3510, 3840
     4, 4270, 4720, 5270, 5930, 6690, 7700, 8920, 10600, 12770/
      DATA K3/347, 358, 369, 381, 394, 409, 425, 442, 460, 479, 500, 522
     1, 544, 568, 596, 622, 652, 685, 718, 753, 795, 835, 879, 924
     2, 976, 1030, 1090, 1160, 1230, 1310, 1410, 1510, 1620, 1740
     3, 1870, 2050, 2240, 2490, 2780, 3120, 3490, 3930, 4530, 5170
     4, 5980, 6980, 8180, 9800, 11830, 14630, 18510/
      DATA (K4(I, 1), I=1, 51)/487, 502, 518, 533, 551, 572, 593, 615
     1, 640, 666, 694, 722, 753, 785, 822, 858, 897, 943, 986, 1030
     2, 1090, 1140, 1200, 1260, 1330, 1400, 1480, 1570, 1670, 1770
     3, 1900, 2030, 2170, 2330, 2510, 2730, 2980, 3290, 3660, 4100
     4, 4570, 5130, 5870, 6690, 7700, 8940, 10400, 12500, 15000
     5, 18400, 23200/
      DATA (K4(I, 2), I=1, 51)/180, 185, 191, 197, 203, 210, 218, 225
     1, 235, 243, 253, 263, 274, 285, 298, 311, 323, 341, 355, 372, 392
     2, 410, 430, 452, 476, 500, 528, 559, 592, 629, 673, 718, 766, 820
     3, 880, 956, 1040, 1150, 1270, 1410, 1570, 1750, 2000, 2260,
     42590, 3000, 3480, 4130, 4930, 6040, 7560/
      DATA CLTOKV, CDGTRD, CMTKF/273. 16, . 0174533, 3. 2808399/
C WS  DATA GG, RE/9. 795168, 6347250. /
      DATA GG, RE/9. 822888, 6370233. /
C
C
      WRITE(5, 450)
      READ(5, 600) NODF
      DO 5 J=1, 60
      IZ=61-J
      IF (LKTIME(IZ). GT. 0) GO TO 6
5     CONTINUE
      IB=1
      GO TO 15
6     AFT=FLOAT(LKTIME(IZ))
      DO 10 J=1, 60
      IZ=61-J
      IF (TTPA(IZ). GE. 0. 0) GO TO 17
10    CONTINUE
      IB=2
15    WRITE(5, 300) IB
      RETURN
17    TM1=LKTIME(1)
      FIZ=IZ
20    IF (TTPA(IZ). GE. TM1) GO TO 25
22    CORTPK(IZ)=999. 9
      TPCOR(IZ)=99. 9
C     ALL 9'S SIGNIFY TIME WAS BEFORE THE FIRST
C     SIGNIFICANT LEVEL WAS CALCULATED.
```

```
      GO TO 45
 25   DO 30 K=1,59
      IF (IZ.GT.51) GO TO 22
      IF (VZA(IZ).LE.-300.0) GO TO 22
      T1=LKTIME(K)
      T2=LKTIME(K+1)
      T3=TTPA(IZ)
      TP1=LKTMP(K)/10.0
      TP2=LKTMP(K+1)/10.0
      IF ((T1.LE.T3).AND.(T2.GE.T3)) GO TO 40
 30   CONTINUE
      GO TO 22
 40   SL=(TP2-TP1)/(T2-T1)
      TPINTC=(SL*(T3-T1))+TP1
      TMPINT=TPINTC+CLTOKV
      TP1=TP1+CLTOKV
      TP2=TP2+CLTOKV
      A=K1(IZ)/(10.0**6)
      B=K2(IZ)/(10.0**3)
      C= K3(IZ)/(10.0**13)
      FL4=1
      IF (NODE.EQ.1) FL4=2
      D=K4(IZ,FL4)/(10.0**3)
      TPCOR(IZ)=(0.0-A*(VZA(IZ)**2))
     1+(B*(TP2-TP1)/(T2-T1))
     2+(C*(TMPINT**4))-(D)
      CORTPK(IZ)=TMPINT+TPCOR(IZ)
 45   IZ=IZ-1
      IF (IZ.EQ.0) GO TO 33
      IF ((TTPA(IZ).GE.0.0).AND.(TTPA(IZ).LE.AET)) GO TO 20
      DO 35 J=1,IZ
 35   TTPA(J)=-1.0
 33   LIZ=IZ+1
      DO 46 J=1,60
      KZ=61-J
      ALTM=(KZ+19.0)*1000.0
 46   GA(KZ)=(GG*RE*ALTM)/(9.8*(RE+ALTM))
      IF (JC.EQ.0) GO TO 37
      IF ((RSHEAR(9).EQ.-1.0).AND.(RSHEAR(10).EQ.-1.0)) GO TO 42
      WRITE(5,475)
      GO TO 47
 42   BLALT=RTIMM(8)
      BLPK=RTIMM(9)
      BLTPC=RTIMM(10)
      BLTMP=BLTPC+CLTOKV
      GO TO 41
 37   RSHEAR(9)=0.0
      RSHEAR(10)=0.0
 47   WRITE(5,500)
      READ(5,600) J
```

```
      IF (J.EQ.1) GO TO 48
44    WRITE(5,700)
      READ(5,800) BLALT,BLPR,BLTMP
      BLTPC=BLTMP
      BLTMP=BLTMP+CLTOKV
      IF ((GA(LIZ)-(BLALT*1000.)).GT.2000.0) GO TO 58
      IB=IFIX(BLALT)-19
      CORTPC=CORTPK(IB)-CLTOKV
      IF (ABS(BLTMP-CORTPK(IB)).LE.2.5) GO TO 43
      WRITE(5,900)
      READ(5,600) J
      IF (J.EQ.1)GO TO 48
      GO TO 44
43    RSHEAR(9)=-1.0
      RSHEAR(10)=-1.0
      BLALT=BLALT*1000.0
      RTIMM(8)=BLALT
      RTIMM(9)=BLPR
      RTIMM(10)=BLTPC
41    BLGPA=(BLALT*GG*RE)/(9.8*(RE+BLALT))
      WRITE(3,125) BLPR,CORTPC,BLTPC,BLALT,BLGPA
      WRITE(3,100)
      WRITE(3,150)
      GO TO 50
C
C    FLAG THAT BASE LEVEL DATA IS NOT THERE.
C    INDICATE THIS HAS HAPPENED BY FILLING IN DUMMY VALUES.
48    WRITE(3,130)
      WRITE(3,100)
      WRITE(3,150)
      DO 49 K=1,60
49    PRSS(K)=1.0E+9
      GO TO 80
50    DO 55 K=LIZ,FIZ
      KZ=FIZ-K+LIZ
      IF (BLALT.GT.GA(KZ)) GO TO 65
55    CONTINUE
      IF ((GA(LIZ)-BLALT).LE.2000.0) GO TO 60
58    WRITE(5,950)
      READ(5,600) J
      IF (J.NE.1) GO TO 44
      NODF=-5
C    FLAG SHOWING BAD BASE LEVEL.
      WRITE(5,400)
      RETURN
60    KZ=LIZ
65    J=KZ
      PRSS(J)=BLPR*EXP((BLALT-GA(J))/
     1  (14.63725*(CORTPK(J)+BLTMP)))
      KZ=J+1
```

```
      DO 70 K=KZ,FIZ
      IF (CORTPK(K).NE.999.9) GO TO 67
      PRSS(K)=1.0E+9
      GO TO 70
67    PRSS(K)=PRSS(K-1)*EXP((GA(K-1)-GA(K))/
     1  (14.63725*(CORTPK(K)+CORTPK(K-1))))
70    CONTINUE
      IF (J.LE.LIZ) GO TO 80
      KZ=J-1
      JZ=KZ-LIZ+1
      DO 75 IZ=1,JZ
      K=KZ+1-IZ
75    PRSS(K)=PRSS(K+1)*EXP((GA(K+1)-GA(K))/
     1  (14.63725*(CORTPK(K+1)+CORTPK(K))))
80    IZ=FIZ
83    IF (PRSS(IZ).NE.1.0E+9) GO TO 85
      DENS(IZ)=0.0
      GO TO 88
85    DENS(IZ)=PRSS(IZ)*348.38/CORTPK(IZ)
88    SPSND(IZ)=20.0514*SQRT(ABS(CORTPK(IZ)))
      IF (CORTPK(IZ).EQ.999.9) SPSND(IZ)=999.9
      ZFA=ZMA(IZ)*CMTKF
      AVDA=(VDA(IZ)-180.0)*CDGTRD
      VXA=VSA(IZ)*SIN(AVDA)
      VYA=VSA(IZ)*COS(AVDA)
      IF (IZ.NE.FIZ) GO TO 90
      VYNA=0.0
      GO TO 95
90    VYNA=SQRT((VXA-AVXA)**2+(VYA-AVYA)**2)
      VYNA=VYNA/1000.0
95    AVXA=VXA
      AVYA=VYA
      CORTPC=CORTPK(IZ)-CLTOKV
      IF (CORTPK(IZ).EQ.999.9) CORTPC=99.9
98    WRITE(3,200) TIPA(IZ),ZMA(IZ),ZFA,VYA,VXA
     1,VSA(IZ),VDA(IZ),VZA(IZ),VYNA,CORTPC
     2,CORTPK(IZ),TPCOR(IZ),PRSS(IZ),DENS(IZ),SPSND(IZ)
      IZ=IZ-1
      IF (IZ.GE.LIZ) GO TO 83
      RETURN
100   FORMAT (1H0,/,4X
     1,'***1 KM LEVEL THERMO DATA PRINTOUT***')
125   FORMAT('BASE LEVEL PRESSURE',F6.2,' MB, ROCKET TEMP'
     1,F7.2,' RAOB TEMP',F7.2,' (DEG-C)',/,'GEOMETRIC ALT'
     2,F7.0,' METERS,     GEOPOTENTIAL ALT',F7.0,' METERS')
130   FORMAT('NO BASE LEVEL DATA')
150   FORMAT ('0    TIME     ALTITUDE(MSL)     WIND VELOCITY(M/SEC)  DIR    FL
     1 VEL WND SHEAR         CORR TEMPS          PRES       DENS      SP SND',/
     2,'     (SEC)    (KM)     (KFT)    -N+S    -E+W    TOTAL   (DEG)   (M/SC)   (
     3MPS/M)   (CENT)   (KEL)    CORR      (MB)     (G/CU M)   (M/S)',/)
```

```
200   FORMAT (1H ,F7.1,2F8.1,3F7.1,F8.1,F7.0,F9.3,2F8.1,F7.1
     1,2X,F7.3,F9.3,F8.1)
300   FORMAT (1H ,'ERROR IN TEMP CORRECTION SUBROUTINE',2X,I1)
400   FORMAT ('  CANNOT COMPUTE PRESSURES DUE TO',/
     1,' DIFFERENCES IN ALTITUDES OF BASE LEVEL',/
     2,' AND FIRST DATA POINT;',/
     3,' BUT WILL CONTINUE.')
450   FORMAT(3X,'DAY OR NIGHT FLIGHT?  DAY-PRESS CR;  NIGHT- 1CR')
475   FORMAT(3X,'BASE LEVEL DATA IS NOT ON MAG TAPE')
500   FORMAT(3X,'IS BASE LEVEL DATA AVAILABLE?'
     1,/,3X,'  YES- PRESS CR;  NO- 1CR')
600   FORMAT(I1)
700   FORMAT(3X,'TYPE IN BASE LEVEL DATA (INCLUDE DECIMAL POINT)'
     1,/,3X,34H  'ALT (KM)'PRSS(MB)'TEMP(DEG-C)'  )
800   FORMAT(5X,F8.0,1X,F8.0,1X,F11.0)
900   FORMAT(3X,'BASE LEVEL TEMP IS NOT WITHIN 2.5 DEGREES'
     1,/,3X,'OF TEMP MEASURED BY LOKI AT SAME ALTITUDE',/,3X
     2,'TRY ANOTHER BASE LEVEL?  YES- PRESS CR;  NO- 1CR')
950   FORMAT(3X,'BASE LEVEL ALT IS MORE THAN 2 KM BELOW',/,3X
     1,'BOTTOM OF LOKI FLIGHT',/,3X
     2,'TRY ANOTHER BASE LEVEL?  YES- PRESS CR;  NO- 1CR')
$ASSM
  LIST
$FORT
      END
```

SUBROUTINE SLML

```
$ASSM
  SCRAT
  NOSOZ
  NORX3
SLAML  PROG  SIG AND MAND LEVELS  26 MAY 77 R02  S- SLML FOR
  NLSTC
  CROSS
  EXTRN PRINTHED, IGAS
  NLIST
$FORT
      SUBROUTINE SLML
C
C    THIS PROGRAM COMES IN AFTER THE 1KM LEVEL THERMO DATA
C    ARRAY (WITH CORRECTED TEMPS) IS CALCULATED.   IT USES
C    THIS DATA TO INTERPOLATE INTO: FIRST FOR SIGNIFICANT
C    LEVELS, AND THEN FOR MANDATORY LEVELS.   ALSO, AFTER
C    EACH LEVEL PRINTOUT IS CALCULATED AND THEN PRINTED,
C    THE MRN ROUTINE IS CALLED TO CALCULATE THAT DATA AND
C    PRINT IT OUT.
C
      COMMON /LKTMTP/ LKTMP,LKTIME
      COMMON /BUFDTA/ RSVR,MRNA
      COMMON /BLOCK2/ OKLV
C   OKLV- 1KM LEVEL DATA
C   OKLV(I,J) J= 1-WS,2-WD,3-CORTEMP(KEL),4-TIME,5-ALT,6-TPCOR
C    7-PRES,8-DEN,9-FL VEL,10-SP SND,11-UNC NS,12-UNC EW
C   RSVR IS ALL OF REST BUFDTA GROUPED TOGETHER
C   MDTA IS SET OF VALUES SENT TO BE MRN FORMATTED
C
      REAL MRNA(60,12),OKLV(60,12),RSVR(11),PC(5),DPC(5)
      REAL MDTA(12)
      INTEGER*2 LKTMP(60),LKTIME(60)
      INTEGER*2 N,I,J,K,ID,IS,IE,IEE,IEX,L,M
      INTEGER*2 NS,NE,NI,JS,JE,JPC,JPCS,IER,IPF,JNOT
      DATA CLTOKV,CMTKF,CDGTRD/273.16,3.2808399,.0174533/
      DATA DPC/70.,50.,30.,20.,10./
      DATA GG,RE/9.822888,6370233./
C WS  DATA GG,RE/9.795168,6347250./
C
C
C   MRN FOR 1KM LEVELS
C
$ASSM
  LHI  7,H'312'
  LE  0,MRNA(7)
  STE  0,PF1
  AIS  7,4
  LE  0,MRNA(7)
  STE  0,PF2
$FORT
```

```
      IPF=0
      IF ((PF1.NE.-1.0).AND.(PF2.NE.-1.0)) GO TO 5
      IPF=1
$ASSM
  LHI  7,H'348'
  LE  0,MRNA(7)
  STE  0,BLALT
  AIS  7,4
  LE  0,MRNA(7)
  STE  0,BLPR
  AIS  7,4
  LE  0,MRNA(7)
  STE  0,BLTPC
$FORT
      BLGPA=(BLALT*GG*RE)/(9.8*(RE+BLALT))
      K=(BLALT/1000.)-19.
      BLTMP=OKLV(K,3)-CLTOKV
5     DO 10 K=1,60
      IF (OKLV(K,4).GE.0.0) GO TO 20
10    CONTINUE
20    IS=K
      DO 30 K=IS,60
      IF (OKLV(K,4).LT.0.0) GO TO 40
30    CONTINUE
      K=61
40    IE=K-1
      ID=30
C
C   MRN FOR SIG LEVELS
C
      DO 60 J=1,60
      K=61-J
      IF (LKTIME(K).GT.0) GO TO 70
60    CONTINUE
      IER=1
65    WRITE (5,900) IER
      STOP
70    NE=K
      NS=1
      T1=LKTIME(NE)
      IF (T1.LT.OKLV(IS,4)) GO TO 90
      DO 75 J=1,60
      K=J
      T1=LKTIME(K)
      IF (T1.GT.OKLV(IS,4)) GO TO 80
75    CONTINUE
80    NE=K-1
90    T1=LKTIME(1)
      IF (OKLV(IE,4).LE.T1) GO TO 98
      DO 93 J=1,60
```

```
        T1=LKTIME(J)
        IF (OKLV(IE,4).LE.T1) GO TO 95
 93     CONTINUE
        IER=11
        GO TO 65
 95     NS=J
 98     IEE=IE-1
        DO 160 N=NS,NE
        NI=N
        DO 120 K=IS,IFE
        T1=LKTIME(N)
        IF (T1.NE.OKLV(K,4)) GO TO 110
        DO 100 I=1,12
 100    MRNA(NI,I)=OKLV(K,I)
        GO TO 160
 110    IF ((OKLV(K,4).GE.T1).AND.(OKLV(K+1,4).LE.T1)) GO TO 130
 120    CONTINUE
        IER=2
        GO TO 65
 130    SLOPE=(T1-OKLV(K,4))/(OKLV(K+1,4)-OKLV(K,4))
        DO 140 I=1,12
        MRNA(NI,I)=OKLV(K,I)+(SLOPE*(OKLV(K+1,I)-OKLV(K,I)))
 140    CONTINUE
        IF(OKLV(K+1,7).NE.1.0E+9) GO TO 160
        MRNA(NI,3)=999.9
        MRNA(NI,6)=99.9
        MRNA(NI,7)=1.0E+9
        MRNA(NI,8)=0.0
        MRNA(NI,10)=999.9
 160    CONTINUE
$ASSM
   BAL  15,PRINTHED
$FORT
        IF (IPF.EQ.1) GO TO 165
        WRITE (3,980)
        GO TO 167
 165    WRITE(3,990) BLPR,BLTMP,BLTPC,BLALT,BLGPA
 167    WRITE(3,920)
        WRITE(3,930)
        DO 190 I=NS,NE
        CORTPC=MRNA(I,3)-CLTOKV
        IF(MRNA(I,3).EQ.999.9) CORTPC=99.9
        ZF=MRNA(I,5)*CMTKF
        AVD=(MRNA(I,2)-180.)*CDGTRD
        VX=MRNA(I,1)*SIN(AVD)
        VY=MRNA(I,1)*COS(AVD)
        IF (I.NE.NS) GO TO 170
        VSH=0.0
        GO TO 180
 170    VSH=SQRT((VX-AVX)**2+(VY-AVY)**2)
```

```
      VSH=VSH/1000.0
180   AVX=VX
      AVY=VY
      WRITE(3,940) MRNA(I,4),MRNA(I,5),ZF,VY,VX,MRNA(I,1)
     1,MRNA(I,2),MRNA(I,9),VSH,CORTPC,MRNA(I,3)
     2,(MRNA(I,J),J=6,8),MRNA(I,10)
190   CONTINUE
$ASSM
 BAL  15,PRINTHED
$FORT
      IF (IPF.EQ.1) GO TO 194
      WRITE(3,980)
      GO TO 197
194   WRITE (3,990) BLPR,BLTMP,BLTPC,BLALT,BLGPA
197   WRITE(3,950)
      NI=NS
      ID=30
      WRITE(3,910)
      I=IE
490   T1=LKTIME(NI)
      IF (T1.LT.OKLV(I,4)) GO TO 510
      DO 500 M=1,12
500   MDTA(M)=OKLV(I,M)
      CALL MRN(ID,MDTA)
      IF(T1.NE.OKLV(I,4)) GO TO 530
      NI=NI+1
      GO TO 530
510   DO 520 M=1,12
520   MDTA(M)=MRNA(NI,M)
      CALL MRN(ID,MDTA)
      NI=NI+1
      GO TO 490
530   I=I-1
      IF(I.GE.IS) GO TO 490
C
C    MRN FOR MANDATORY LEVELS
C
      IF (IPF.EQ.1) GO TO 200
      GO TO 390
200   DO 210 K=1,5
210   PC(K)=DPC(K)
      JNOT=0
$ASSM
  LIS  7,1
  STH  7,IGAS
$FORT
      DO 240 JPC=1,10
      DO 220 K=1,5
      IF(PC(K).NE.0.3) GO TO 215
      IF(OKLV(IS,7).LT.0.4) GO TO 215
```

```
        K=K-1
        PC(K)=0.4
        GO TO 250
  215   IF (OKLV(IS,7).GE.PC(K)) GO TO 250
  220   CONTINUE
        DO 230 K=1,5
  230   PC(K)=PC(K)/10.
  240   CONTINUE
        IER=3
        GO TO 65
  250   JPCS=K
        J=60
        DO 320 JPC=1,10
        DO 300 K=JPCS,5
        IF(JNOT.EQ.1) GO TO 251
        IF(PC(K).NE.0.3) GO TO 251
        JNOT=1
        K=K-1
        PC(K)=0.4
  251   IEX=IE
        IF(IE.GT.51) IEX=51
        IF (OKLV(IEX,7).GT.PC(K)) GO TO 330
        DO 260 I=IS,IEE
        IF (OKLV(I,7).NE.PC(K)) GO TO 255
        DO 253 M=1,12
  253   MRNA(J,M)=OKLV(I,M)
        GO TO 290
  255   IF ((OKLV(I,7).GE.PC(K)).AND.(OKLV(I+1,7).LE.PC(K)))
       1 GO TO 270
  260   CONTINUE
        IER=4
        GO TO 65
  270   SLOPE=(PC(K)-OKLV(I,7))/(OKLV(I+1,7)-OKLV(I,7))
        DO 280 M=1,12
        MRNA(J,M)=OKLV(I,M)+(SLOPE*(OKLV(I+1,M)-OKLV(I,M)))
  280   CONTINUE
        IF(OKLV(I+1,7).NE.1.0E+9) GO TO 290
        MRNA(J,3)=999.9
        MRNA(J,6)=99.9
        MRNA(J,7)=1.0E+9
        MRNA(J,8)=0.0
        MRNA(J,10)=999.9
  290   J=J-1
        JPCS=1
  300   CONTINUE
        IF(PC(2).NE.0.4) GO TO 305
        PC(2)=0.5
  305   DO 310 M=1,5
  310   PC(M)=PC(M)/10.
  320   CONTINUE
```

```
          IER=5
          GO TO 65
   330    CONTINUE
          JE=60
          JS=J+1
$ASSM
     BAL  15,PRINTHED
$FORT
          WRITE(3,990) BLPR,BLTMP,BLTPC,BLALT,BLGPA
          WRITE(3,960)
          WRITE(3,930)
          DO 360 J=JS,JE
          CORTPC=MRNA(J,3)-CLTOKV
          IF(MRNA(J,3).EQ.999.9) CORTPC=99.9
          ZF=MRNA(J,5)*CMTKF
          AVD=(MRNA(J,2)-180.0)*CDGTRD
          VX=MRNA(J,1)*SIN(AVD)
          VY=MRNA(J,1)*COS(AVD)
          IF(J.NE.JS) GO TO 340
          VSH=0.0
          GO TO 350
   340    VSH=SQRT((VX-AVX)**2+(VY-AVY)**2)
          VSH=VSH/1000.0
   350    AVX=VX
          AVY=VY
          WRITE(3,940) MRNA(J,4),MRNA(J,5),ZF,VY,VX,MRNA(J,1)
         1,MRNA(J,2),MRNA(J,9),VSH,CORTPC,MRNA(J,3)
         2,(MRNA(J,I),I=6,8),MRNA(J,10)
   360    CONTINUE
$ASSM
     BAL  15,PRINTHED
$FORT
          WRITE(3,990) BLPR,BLTMP,BLTPC,BLALT,BLGPA
          WRITE(3,970)
          ID=40
          WRITE(3,910)
          DO 380 J=JS,JE
          DO 370 M=1,12
   370    MDTA(M)=MRNA(J,M)
          CALL MRN(ID,MDTA)
   380    CONTINUE
   390    CONTINUE
$ASSM
     BAL  15,PRINTHED
$FORT
          IF(IPF.EQ.1) GO TO 400
          WRITE(3,980)
          GO TO 410
   400    WRITE(3,990) BLPR,BLTMP,BLTPC,BLALT,BLGPA
   410    RETURN
```

```
900   FORMAT(' ERROR IN SLML; IER=',I2)
910   FORMAT(1H0,5X,'STA  DATE TIME ALT DIR SP UNS UEW'
     1,1X,'NS  EW FV TMP COR  PRES   DEN  S/S')
920   FORMAT(1H0,4X,'***SIGNIFICANT LEVEL THERMO DATA***')
930   FORMAT('0   TIME   ALTITUDE(MSL)   WIND VELOCITY(M/SEC)'
     2,2X,'DIR   FL VEL WIND SHEAR        CORR TEMPS'
     3,8X,'PRES      DENS     SP SND',/,'     (SEC)    (KM)'
     4,3X,'(KFT)    -N+S    -E+W    TOTAL  (DEG)  (M/SC)  (MPS/M)'
     5,2X,'(CENT)  (KEL)   CORR     (MB)    (G/CU M)  (M/S)',/)
940   FORMAT(1H ,F7.1,2F8.1,3F7.1,F8.1,F7.0,F9.3,2F8.1,F7.1
     1,2X,F7.3,F9.3,F8.1)
950   FORMAT(1H0,6X,'***MRN - 30 CARDS***')
960   FORMAT(1H0,4X,'***MANDATORY LEVEL THERMO DATA***')
970   FORMAT(1H0,6X,'***MRN - 40 CARDS***')
980   FORMAT('NO BASE LEVEL DATA')
990   FORMAT('BASE LEVEL PRESSURE',F6.2,' MB, ROCKET TEMP'
     1,F7.2,' RAOB TEMP',F7.2,' (DEG-C)',/,'GEOMETRIC ALT'
     2,F7.0,' METERS,     GEOPOTENTIAL ALT',F7.0,' METERS')
$ASSM
  LIST
$FORT
      END
```

SUBROUTINE MRN

```
$ASSM
  SCRAT
  NOSQ2
  NORX3
MRNMSG   PROG     MRN MESSAGE     17 MAY 77   R01   S - MRNFL.FOR
  NLSTC
  CROSS
  NLIST
$FORT
      SUBROUTINE MRN(ID,MRNA)
C
C   THIS ROUTINE GENERATES A MRN-FORMATTED MESSAGE WHEN
C   GIVEN ONE LINE OF ATMOSPHERIC DATA:
C     30 CARDS - 1 KM LEVEL THERMO DATA OR SIG LEVEL DATA
C     40 CARDS - MANDATORY (CONSTANT PRESSURE) LEVEL DATA
C   ID IS IDENTIFICATION (=30 OR 40)
C   IS AND IE ARE THE LIMITS ON DATA ARRAY
C
      COMMON /BUFDTA/RSVR,ALRST
      REAL RSVR(11),ALRST(720)
      REAL MRNA(12),KVTOCL
C    MRNA(J)   J= 1-WS,2-WD,3-CORTEMP(K),4-TIME,
C      5-ALT,6-TPCOR,7-PRES,8-DENS,9-FL VEL,10-SP SND
C      11-UNC NS,12-UNC EW
      INTEGER*2 MK(39),TMK,ID,IS,IE,ISD(4),I,K,JMK,JSD(9)
      DATA KVTOCL,CDGTRD/273.16,.0174533/
C WS  DATA GG,RE/9.795168,6347250./
      DATA GG,RE/9.822888,6370233./
C
C
C WS  ISTA1=72269
C   STATION NUMBERS
      ISTA1=70192
$ASSM
   LIS   3,4
   LIS   4,0
   LHI   5,H'20'
AGN   LH    6,RSVR(5)
   STH   6,ISD(4)
   AIS   4,2
   AIS   5,2
   SIS   3,1
   BNZ   AGN
$FORT
      CONTINUE
      JSD(1)=ISD(4)
      JSD(2)=ISD(2)/10
      JSD(3)=ISD(2)-JSD(2)*10
      JSD(4)=ISD(3)/10
      JSD(5)=ISD(3)-JSD(4)*10
```

```
      JSD(6)=ISD(1)/1000
      JMK=ISD(1)-JSD(6)*1000
      JSD(7)=JMK/100
      JMK=JMK-JSD(7)*100
      JSD(8)=JMK/10
      JSD(9)=JMK-JSD(8)*10
C    ALTITUDE
      IF (ID.EQ.30) GO TO 5
      ATMK=(GG*RE*MRNA(5))/(9.8*(RE+MRNA(5)))
      TMK=INT((ATMK*100.0)+.5)
      GO TO 6
5     TMK=INT((MRNA(5)*100.0)+.5)
6     MK(1)=TMK/10000
      MK(2)=TMK-MK(1)*10000
C    WIND DIRECTION
      TMK=INT(MRNA(2)+.5)
      MK(3)=TMK/100
      TMK=TMK-MK(3)*100
      MK(4)=TMK/10
      MK(5)=TMK-MK(4)*10
C    WIND SPEED
      TMK=INT(MRNA(1)+.5)
      MK(6)=TMK/100
      TMK=TMK-MK(6)*100
      MK(7)=TMK/10
      MK(8)=TMK-MK(7)*10
      IF (MK(8).NE.0) GO TO 10
      MK(3)=0
      MK(4)=0
      MK(5)=0
10    AWD=(MRNA(2)-180.0)*CDGTRD
C    UNCORRECTED NORTH SOUTH WIND
      TMK=INT(ABS(MRNA(11))+.5)
      MK(9)=' '
      IF (MRNA(11).LT.-0.5) MK(9)='-'
      MK(10)=TMK/100
      TMK=TMK-MK(10)*100
      MK(11)=TMK/10
      MK(12)=TMK-MK(11)*10
C    UNCORRECTED EAST WEST WIND
      TMK=INT(ABS(MRNA(12))+.5)
      MK(13)=' '
      IF (MRNA(12).LT.-0.5) MK(13)='-'
      MK(14)=TMK/100
      TMK=TMK-MK(14)*100
      MK(15)=TMK/10
      MK(16)=TMK-MK(15)*10
C    NORTH SOUTH WIND
      ATMK=MRNA(1)*COS(AWD)
      TMK=INT(ABS(ATMK)+.5)
```

```
      MK(17)=' '
      IF (ATMK.LT.-0.5) MK(17)='-'
      MK(18)=TMK/100
      TMK=TMK-MK(18)*100
      MK(19)=TMK/10
      MK(20)=TMK-MK(19)*10
C   EAST WEST WIND
      ATMK=MRNA(1)*SIN(AND)
      TMK=INT(ABS(ATMK)+.5)
      MK(21)=' '
      IF (ATMK.LT.-0.5) MK(21)='-'
      MK(22)=TMK/100
      TMK=TMK-MK(22)*100
      MK(23)=TMK/10
      MK(24)=TMK-MK(23)*10
C   FALL VELOCITY
      TMK=INT(ABS(MRNA(9))+.5)
      MK(25)=TMK/100
      TMK=TMK-MK(25)*100
      MK(26)=TMK/10
      MK(27)=TMK-MK(26)*10
      IF (MRNA(3).LT.999.9) GO TO 30
      MK(28)='+'
      DO 20 K=29,31
20    MK(K)=9
      MK(32)='+'
      MK(33)=9
      MK(34)=9
      GO 10 40
C   CORRECIED TEMP(C)
30    ATMK=MRNA(3)-KVTOCL
      TMK=INT(ABS(ATMK)+.5)
      MK(28)=' '
      IF (ATMK.LT.-0.5) MK(28)='-'
      MK(29)=TMK/100
      TMK=TMK-MK(29)*100
      MK(30)=TMK/10
      MK(31)=TMK-MK(30)*10
C   TEMPERATURE CORRECTION
      TMK=INT(ABS(MRNA(6))+.5)
      MK(32)=' '
      IF (MRNA(6).LT.-0.5) MK(32)='-'
      MK(33)=TMK/10
      MK(34)=TMK-MK(33)*10
40    IF (MRNA(7).LT. 1.0E+9) GO TO 50
      F1=9.999
      F2=9.999
      MK(35)='+'
      MK(36)=9
      MK(37)='+'
```

```
      MK(38)=9
      MK(39)=999
      GO TO 65
C   PRESSURE
50    ATMK=ALOG10(MRNA(7)+.000001)
      IF (ATMK.LT.0.) ATMK=ATMK-1.0
      MK(35)='+'
      IF (ATMK.LT.0.) MK(35)='-'
      MK(36)=INT(ATMK)
      F1=MRNA(7)/(10.0**MK(36))
      MK(36)=IABS(MK(36))
C   DENSITY
      ATMK=ALOG10(MRNA(8)+.000001)
      IF (ATMK.LT.0.) ATMK=ATMK-1.0
      MK(37)='+'
      IF (ATMK.LT.0.) MK(37)='-'
      MK(38)=INT(ATMK)
      F2=MRNA(8)/(10.0**MK(38))
      MK(38)=IABS(MK(38))
C   SPEED OF SOUND
60    MK(39)=INT(MRNA(10)+.5)
65    IF (ID.EQ.40) GO TO 70
      WRITE(3,300) ISTAT,(JSD(K),K=1,9)
     1,(MK(K),K=1,34),F1,MK(35),MK(36),F2,MK(37)
     2,MK(38),MK(39),ID
      GO TO 100
70    WRITE(3,350) ISTAT,(JSD(K),K=1,9)
     1,(MK(K),K=1,24),(MK(K),K=28,34),F1,MK(35),MK(36)
     2,F2,MK(37),MK(38),MK(39),ID
100   CONTINUE
      RETURN
300   FORMAT(1H ,4X,I5,I2,8I1,I1,I4,6I1,3(A1,3I1),A1
     1,6I1,A1,3I1,A1,2I1,2(F5.3,A1,I1),I3,9X,I2)
350   FORMAT(1H ,4X,I5,I2,8I1,I1,I4,6I1,4(A1,3I1)
     1,3X,A1,3I1,A1,2I1,2(F5.3,A1,I1),I3,9X,I2)
$ASSM
  LIST
$FORT
      END
```

SUBROUTINE XYPLOT

```
$ASSM
  SCRAT
  NOSOZ
  NORX3
XYTPLOT  PROG  X&Y WIND, TEMP PLOT  18 AUG 77 R02  S- XYPLOT.FOR
  NLSTC
  CROSS
  NLIST
$FORT
      SUBROUTINE XYPLOT
C
C   THIS PROGRAM ACCEPTS THE 1-KM LEVEL THERMO DATA ARRAY,
C   AND CONSTRUCTS A GRAPH FROM THE DATA.  X AND Y WINDS
C   AND CORRECTED TEMPERATURE ARE PLOTTED VS ALTIDUDE.
C
      COMMON /BLOCK2/OKLV
      DIMENSION OKLV(60,12)
      INTEGER*2 JV(100),K,IS,IDX,IDY,IDT,ISH,J
      DATA CDGTRD,CLTOKV/.0174533,273.16/
C
C
      WRITE(3,900)
      WRITE(3,910)
      WRITE(3,920)
      WRITE(3,930)
      DO 10 K=1,60
      IF(OKLV(K,4).GE.0.0) GO TO 20
10    CONTINUE
20    IS=K
      DO 30 K=IS,60
      IF(OKLV(K,4).LT.0.0) GO TO 40
30    CONTINUE
      K=61
40    IE=K-1
      JV(100)='I'
C   START AT TOP
      K=IE
50    AVD=(OKLV(K,2)-180.)*CDGTRD
      VX=OKLV(K,1)*SIN(AVD)
      VY=OKLV(K,1)*COS(AVD)
      TPC=OKLV(K,3)-CLTOKV
      DO 75 J=1,99
75    JV(J)=' '
      IDX=(VX/5.0)+50.5
      IDY=(VY/5.0)+50.5
      IDT=(TPC/2.5)+90.5
      IF((IDX.LT.0).OR.(IDX.GE.100)) GO TO 150
      IF((IDY.LT.0).OR.(IDY.GE.100)) GO TO 150
      IF(IDT.LT.100) GO TO 100
      IF(IDX.NE.IDY) GO TO 90
```

```
      JV(IDX)='O'
      GO TO 150
90    JV(IDX)='X'
      JV(IDY)='Y'
      GO TO 150
100   IF((IDX.NE.IDY).OR.(IDX.NE.IDT).OR
     1.(IDY.NE.IDT)) GO TO 115
      JV(IDX)='O'
      GO TO 150
115   IF(IDX.NE.IDY) GO TO 120
      JV(IDX)='O'
      JV(IDT)='T'
      GO TO 150
120   IF(IDY.NE.IDT) GO TO 130
      JV(IDY)='O'
      JV(IDX)='X'
      GO TO 150
130   IF(IDX.NE.IDT) GO TO 140
      JV(IDX)='O'
      JV(IDY)='Y'
      GO TO 150
140   JV(IDX)='X'
      JV(IDY)='Y'
      JV(IDT)='T'
150   ALT=OKLV(K,5)
      WRITE(3,940) ALT,(JV(J),J=1,100),ALT
      K=K-1
      IF(K.GE.IS) GO TO 50
      WRITE(3,930)
      WRITE(3,920)
      WRITE(3,910)
      RETURN
900   FORMAT(1H0,4X
     1,'*** X AND Y WIND(M/S) AND TEMP(DEG-C) . VS. ALT(KM) ***',//)
910   FORMAT(68X,'-75',7X,'-50',7X,'-25',8X,'0',8X,'25')
920   FORMAT(7X,'-250',6X,'-200',6X,'-150',6X,'-100',7X,'-50'
     1,8X,'0',9X,'50',7X,'100',7X,'150',7X,'200',7X,'250')
930   FORMAT(9X,10('I*********'),'I')
940   FORMAT(2X,F5.2,2X,'I',100A1,2X,F5.2)
$ASSM
  LIST
$FORT
      END
```

SUBROUTINE ROCOB

```
$ASSM
   SCRAT
   NOSQZ
   NORX3
ROCOBMSG   PROG   ROCOB MESSAGE   26 APR 77  K01   S - ROCOBFL.FOR
   NLSTC
   CROSS
   NLIST
$FURT
C
      SUBROUTINE ROCOB
C
C    THIS ROUTINE CALCULATES THE ROCOB MESSAGE.  IT PRINTS
C    OUT ALL LEVELS SELECTED AND THE REASON(S) FOR
C    SELECTION; AND THEN THE ROCOB IN CORRECT FORMAT.
C
C
      COMMON /BLOCK2/ROCOBA
      DIMENSION ROCOBA(60,12)
C    ROCOBA(I,J) J= 1-WS,2-WD,3-CORTEMP(K),4-TIME,
C      5-ALT,6-TCOR,7-PRES,8-DEN,9-FL VEL,
C      10-SP SND,11-UNC NS,12-UNC EW
      DIMENSION RPRINT(3)
      INTEGER*2 RSEL(3),IX(12),K,ISTART,IEND
      INTEGER*2 I1,IS,IE,I,IK,IW,IY,IL,J,ISL,IZ,IT
      INTEGER*2 N,IB2,IB3,IB4,IG,I5
C
C
      DO 300 K=1,51
      IF (ROCOBA(K,4).GT.0.) GO TO 310
300   CONTINUE
310   ISTART=K
      DO 320 K=ISTART,51
      IF (ROCOBA(K,4).LT.0.) GO TO 330
320   CONTINUE
      K=52
330   IEND=K-1
C    BEGIN MANDITORY AND SIG LEVELS
      IG=0
      N=0
      IS=ISTART
      IE=IEND
5     DO 10 I=IS,IE
      IK=I+19
      IF ((IK/5.).EQ.FLOAT(IK/5)) GO TO 15
C    LEVELS OF 5KM MANDITORY
      IF (I.EQ.IEND) GO TO 15
10    CONTINUE
      GO TO 180
15    IF ((IS+1)-I) 18,16,10
```

```
16    IS=I
      GO TO 5
18    IS=I
20    DO 40 I1=1,3
      IF ((I-IS).LE.1) GO TO 50
      IF (I1.NE.3) GO TO 25
      IF ((ROCOBA(IS,3).EQ.999.9).OR.(ROCOBA(I,3).EQ.999.9))
     1  GO TO 40
C   CALCULATE STRAIGHT LINES
25    Y1=IS+19
      X1=ABS(ROCOBA(IS,I1))
      Y2=I+19
      X2=ABS(ROCOBA(I,I1))
      XM=(Y1-Y2)/(X1-X2)
      B=Y2-XM*X2
      IW=IS+1
      IY=I-1
      DO 30 IL=IW,IY
      X=(IL+19-B)/XM
C   CHECK FOR SIGNIFICANT LEVELS
      TE1=ABS(X-ABS(ROCOBA(IL,I1)))
      IF ((TE1.GE.5.).AND.(I1.EQ.1)) GO TO 70
      IF ((TE1.GE.3.).AND.(I1.EQ.3)) GO TO 70
      IF (I1.NE.2) GO TO 30
      TE2=ABS(ROCOBA(IL,1))
      IF ((TE1.GE.60.).AND.((TE2.GE.8.).AND
     1 (TE2.LE.15.))) GO TO 70
      IF ((TE1.GE.30.).AND.((TE2.GE.16.).AND
     1 (TE2.LE.30.))) GO TO 70
      IF ((TE1.GE.20.).AND.(TE2.GE.31.)) GO TO 70
30    CONTINUE
40    CONTINUE
50    IS=I
      IF (N.EQ.0) GO TO 5
      IF (N.NE.1) GO TO 55
      I=I5
      N=0
      GO TO 20
55    IF (N.NE.2) GO TO 60
      I=IG
      N=1
      GO TO 20
60    IF (N.NE.3) GO TO 65
      I=IB2
      N=2
      GO TO 20
65    IF (N.NE.4) GO TO 275
      I=IB3
      N=3
      GO TO 20
```

```
C   IF SIGNIFICANT LEVEL GENERATED - MAKE PRAMETER NEGATIVE
C   THAT CAUSED THE LEVEL TO BE SELECTED
70    ROCOBA(IL, I1)= -(ABS(ROCOBA(IL, I1)))
      I=IL
      IF (N.NE.0) GO TO 75
      IG=IL
      N=1
      GO TO 20
75    IF (N.NE.1) GO TO 80
      IB2=IL
      N=2
      GO TO 20
80    IF (N.NE.2) GO TO 85
      IB3=IL
      N=3
      GO TO 20
85    IF (N.NE.4) GO TO 275
      IB4=IL
      N=4
      GO TO 20
180   CONTINUE
      WRITE(3,185)
      WRITE(3,190)
C   MARK PRAMETER THAT DETERMINED SIG LEVEL
      DO 250 I=ISTART, IEND
      ISL=0
      DO 230 J=1,3
      VA=ROCOBA(I,J)
      RSEL(J)=' '
      RPRINT(J)=VA
      IF (VA.GT.0.) GO TO 230
      RSEL(J)='X'
      ISL=1
      RPRINT(J)=-VA
230   CONTINUE
      IZ=I+19
      IF ((I.EQ.ISTART).OR.(I.EQ.IEND)
     1.OR.((IZ/5.).EQ.FLOAT(IZ/5))) GO TO 245
      IF (ISL.NE.1) GO TO 250
245   RPRINT(3)=RPRINT(3)-273.16
C   SET UP TO PRINT ROCOR
      IX(1)=IZ
      IF (ABS(ROCOBA(I,3)).LT.999.9) GO TO 246
      IX(2)=9
      IX(3)=9
      IX(4)=9
      RPRINT(3)=999.9
      GO TO 247
246   IX(2)=5
      IT=(ABS(RPRINT(3))+.5)
```

```
      IX(3)=IT/10
      IX(4)=IT-IX(3)*10
      IF (RPRINT(3).GE.0.) IX(2)=0
247   IT=(RPRINT(2)/10.+0.5)
      IF(IT.EQ.0) IT=36
      IX(5)=IT/10
      IX(6)=IT-IX(5)*10
      IT=(RPRINT(1)+.5)
      IX(7)=IT/100
      IT=IT-IX(7)*100
      IX(8)=IT/10
      IX(9)=IT-IX(8)*10
      IX(10)=9
      IF (ROCOBA(I,7).NE.1.0E+9) GO TO 248
      IX(11)=9
      IX(12)=999
      GO TO 249
248   TE3=ROCOBA(I,8)
      IX(11)=5
      IF (TE3.GE.100.0) IX(11)=0
      IF ((TE3.LT.100.).AND.(TE3.GE.10.)) IX(11)=1
      IF ((TE3.LT.10.).AND.(TE3.GE.1.)) IX(11)=2
      IF ((TE3.LT.1.).AND.(TE3.GE.0.1)) IX(11)=3
      IF ((TE3.LT.0.1).AND.(TE3.GE.0.01)) IX(11)=4
      IF (IX(11).EQ.1) TE3=TE3*10
      IF (IX(11).EQ.2) TE3=TE3*100
      IF (IX(11).EQ.3) TE3=TE3*1000
      IF (IX(11).EQ.4) TE3=TE3*10000
      IF (IX(11).EQ.5) TE3=TE3*100000
      IX(12)=INT(TE3+.5)
C    PRINT OUT ALL LEVELS AND REASON(S), AND ROCOB
249   WRITE(3,210) (I2,(RPRINT(K),K=1,3),(RSEL(J),J=1,3)
     1,(IX(N),N=1,12))
250   CONTINUE
      RETURN
275   WRITE(5,600)
      RETURN
185   FORMAT(1H0,6X,'***ROCOB MESSAGE***')
190   FORMAT('0',9X,'KM  WIND SPEED   WIND DIR   TEMP'
     1,5X,'WS   WD   TEMP',19X,'ROCOB')
210   FORMAT(10X,I2,3(5X,F5.1),3(5X,A1),10X,I2,3I1
     1,5X,5I1,5X,2I1,I3)
600   FORMAT (3X,'LOOPING ERROR IN ROCOB ROUTINE')
$ASSM
   LIST
$FORT
      END
```

VI. INTERDATA MODEL 70 STUDY

The Interdata model 70 computer is part of a system called EPAMS(Experimental Prototype Automated Meteorological System). EPAMS is to provide a system to automatically collect met data, process this data and provide users with appropriate meteorological outputs.

Since the Interdata model 70 computer is limited to its present memory capacity of 64K bytes, only computer programs that are smaller than about 50K bytes can be executed(RTOS operating system uses about 16K bytes). This memory limitation places rather serious restrictions on the type and number of programs that can be run. For example the computer program at the beginning of this report utilizes 92K bytes of memory.

With the above as background, the following items were considered in recommending a replacement for the model 70 computer.

1 - Compatibility with existing hardware
2 - Computing capacity memory size and computing speed
3 - Man hours of time to learn new system procedures
4 - Cost

When all four items are considered together, there is only one logical choice. This choice would be the Interdata 7/32 computer. If the cost factor was not present then the computing speed could be enhanced by selecting the 8/32 computer.

The responses to the above areas are as follows:

1 - Compatibility with existing hardware--Since Interdata utilizes the same type of controls for both the model 70 and 7/32 systems, both the controller and I/O devices can be utilized directly by the 7/32 computer. Therefore there is a 100% compatibility with existing hardware.

2 - Computing capacity - memory size and computing speed. The Interdata 7/32 can be expanded to 1 million bytes of memory. Computing speed is 1 to 3 microseconds for most operations. A high performance floating point unit is available which will provide double precision speeds of 5 to 10 microseconds.

3 - The MT(multitasking) operating system available for the 7/32 is structured in a manner which is very similar to the RTOS system of the model 70. It is estimated that a period of 2 to 3 weeks would be required to be able to use the new system for personnel.

4 - Cost - The basic cost of a 7/32 with a MAC(memory access controller), 128K bytes of memory, power supply and card cage is about $15,000. If an 8/32 were to be purchased, the cost is about 3 times that of a 7/32. Also the present MT software is not written to take advantage of the special hardware capabilities that the 8/32 has.

VII. ALASKA EFFORT

From July 12, 1977, through July 24, 1977, Dr. Merrill was at the Poker Flat facility installing and testing the programs developed.

During this period of time, the personnel at Poker Flat were required to launch a balloon and rocket daily (for scale of motion study). The computer was operated utilizing the programs developed. The first four flights were evaluated and checked for accuracy. The results of this evaluation indicated that the program worked very well and was able to get useful data for some flights for which manual reduction failed.

During the remainder of the time, spot checks were made on flight data along with an intensive education program for the Poker Flat personnel so that they would be able to operate the computer with the new programs

☆ U.S. GOVERNMENT PRINTING OFFICE: 1977 777-093/25

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

Field	Entry
1. REPORT NUMBER	ASL-CR-78-0115-1
2. GOVT ACCESSION NO.	ERADCOM/ASL
3. RECIPIENT'S CATALOG NUMBER	CR-78-0115-1
4. TITLE (and Subtitle)	DEVELOPMENT OF A REAL-TIME ROCKETSONDE AND A REAL-TIME RADIOSONDE COMPUTER PROGRAM.
5. TYPE OF REPORT & PERIOD COVERED	Contractor Report.
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s)	M. Don Merrill Scott Fry
8. CONTRACT OR GRANT NUMBER(s)	DAAD07-76-C-0115
9. PERFORMING ORGANIZATION NAME AND ADDRESS	Electrical and Computer Engineering New Mexico State University Las Cruces, New Mexico 88003
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	DA Task No. 1P665702D127-03
11. CONTROLLING OFFICE NAME AND ADDRESS	US Army Electronics Research and Development Command Adelphi, MD 20783
12. REPORT DATE	Jan [redacted] 78
13. NUMBER OF PAGES	134
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	Atmospheric Sciences Laboratory White Sands Missile Range, New Mexico 88002
15. SECURITY CLASS. (of this report)	UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Contract Monitor: Bruce Kennedy

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Rocketsonde
Computer program
Meteorological data reduction
Meteorological rocket network

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains a detailed description of a computer program that was developed for use with the Interdata 7/32 computer and the interfaced Nike Hercules radar systems located at the MTTR site at White Sands, New Mexico and the Poker Flat site in Alaska. → next page

The program contains both assembler and FORTRAN language instructions. The assembler language instructions are used to provide input/output control for data flow from the radar to the computer and from the computer

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Cont)

to the T. V. display. The FORTRAN statements comprise 95% of the program and are used to compute most of the processed data.

The program can process in real-time a complete data reduction for an MRN rocketsonde or a list of 2 minute layer winds for a radiosonde. For a rocketsonde, the program utilizes the temperature telemetry data, the positional radar data, and the operator inputs of rocketsonde temperature calibration values and base level tie-in data from a radiosonde flight to produce an MRN(WDC-A) format listing of the following:

1. 1 KM corrected and uncorrected winds,
2. Significant level temperature data,
3. 1 KM thermodynamic data,
4. Significant level thermodynamic data,
5. MRN 30 cards(image),
6. Mandatory thermodynamic data,
7. MRN 40 cards(image), and
8. Printer plot of X and Y component winds and temperatures versus 1 KM altitudes

For a radiosonde, the program utilizes the positional radar data to produce a listing of two-minute layer winds at 1 minute intervals.

This report includes a description, listing and flow charts for the main program and all subroutines, together with instructions and examples on how to use the program. Also included is the typical output listing for both the radiosonde and rocketsonde flights.