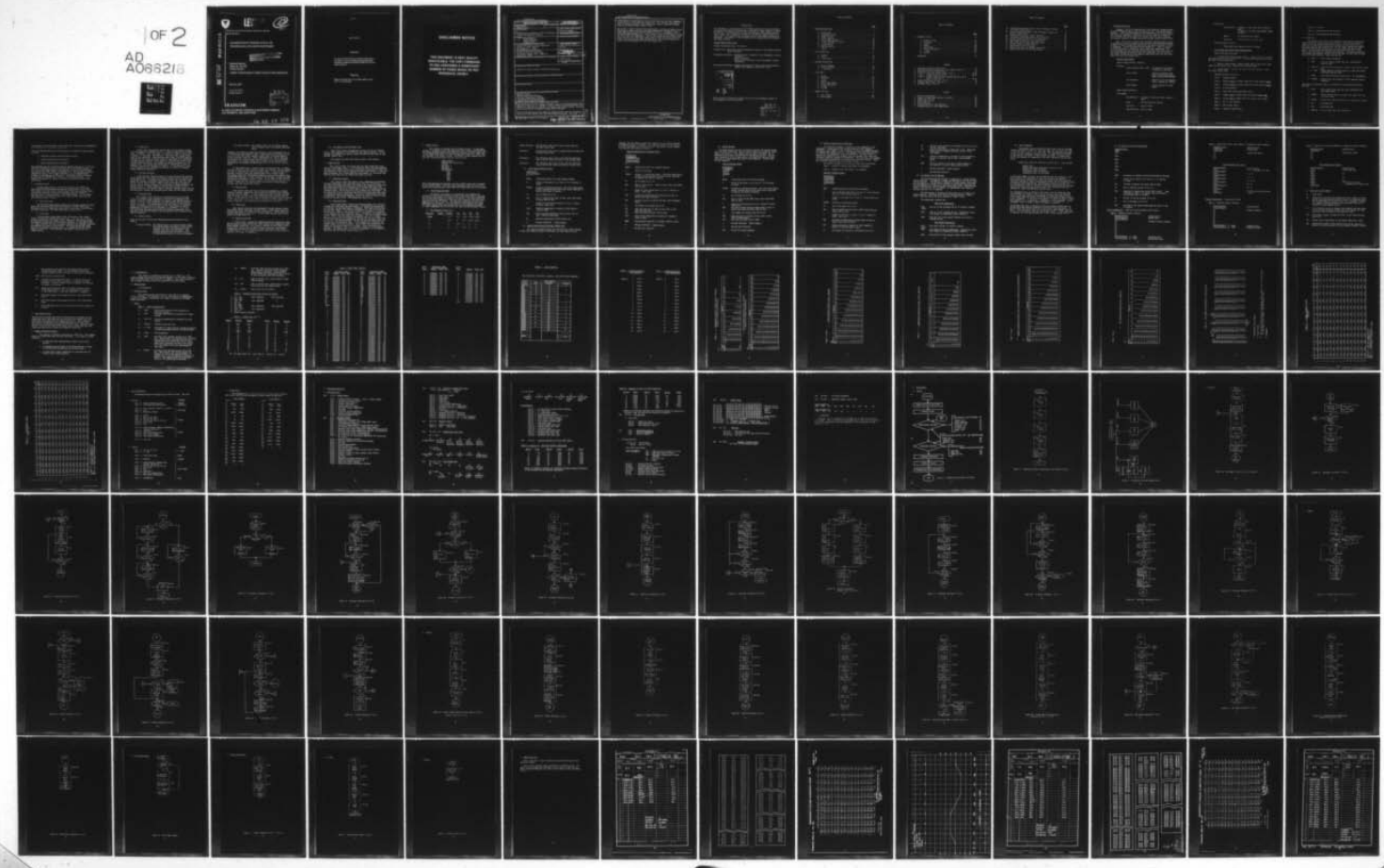
AD-A066 218 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND FO--ETC F/G 4/2
DOCUMENTATION OF SOFTWARE IN THE OL-192 METEOROLOGICAL DATA RED--ETC(U)
FEB 79 R BELLUCCI, S W BURNETT, T RICHTER
UNCLASSIFIED DELCS-TR-79-1 NL

1 OF 2
AD
A066218



LEVEL II

12

NW

RESEARCH AND DEVELOPMENT TECHNICAL REPORT

DELCS-TR-79-1

AD A0 66218

DOCUMENTATION OF SOFTWARE IN THE OL-192 METEOROLOGICAL DATA REDUCTION PROGRAM

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

Raymond Bellucci
Steven W. Burnett
Thomas Richter

COMBAT SURVEILLANCE & TARGET ACQUISITION LABORATORY

DDC FILE COPY

February 1979

DISTRIBUTION STATEMENT

Approved for public release;
distribution unlimited.

DDC
RECEIVED
MAR 22 1979
D

ERADCOM

US ARMY ELECTRONICS RESEARCH & DEVELOPMENT COMMAND
FORT MONMOUTH, NEW JERSEY 07703

79 03 19 076

NOTICES

Disclaimers

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.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 DELCS-TR-79-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Documentation of Software in the OL-192 Meteorological Data Reduction Program,		5. TYPE OF REPORT & PERIOD COVERED 9 Technical Report,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Raymond Bellucci, Steven W. Burnett Thomas Richter		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Combat Surveillance & Target Acquisition Laboratory, ATTN: DELCS-S Fort Monmouth, NJ 07703		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 6945DR.235.80.84.01 QGLZZ-9 ORL 9934
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Electronics Research and Development Command, ATTN: DRDEL-PAO Adelphi, MD 20783		12. REPORT DATE 11 Feb[redacted]79
		13. NUMBER OF PAGES ~~178~~ 12 185 p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		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

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

Computer Program
Meteorological Data Reduction
Ballistic Meteorological Message
Air Weather Meteorological Message
Computer Meteorological Messages

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

This report describes the software contained in the OL-192 Meteorological Data Reduction Program, and is intended to supply complete program description, file definitions, variable descriptions, flow charts, and program listing. The program was written for the Hewlett Packard 9825A Calculator.

The OL-192 Meteorological Data Reduction Program quickly converts raw meteorological data into accurate real time meteorological messages. Real time

CONT

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387389

79 03 13 076

CONT'

meteorological data produced by the OL-192 increase the artillery commander's capability for a first round hit probability. This increase the effectiveness of friendly artillery, while reducing our weapons' exposure to enemy acquisition systems counter battery fires.

The OL-192 is used by the artillery meteorological sections to reduce the raw meteorological data received by the Radiosonde Recorder AN/TMQ-5(*) and the winds data received by the Control Recorder C-577(*), a component of the Radiosonde System AN/GMD-1(*). Input is entered from the calculator's console for real flight time computation. The OL-192 program can also calculate messages from the last flight data, punch pressure calibration tapes for radiosondes, and can print the flight information for each entered significant level and zone information for each zone height level.

PURPOSE

This document is a description of the software contained in the OL-192 Meteorological Data Reduction Program and is intended to supply complete program description, file definitions, variable descriptions, flow charts, and program listing. This program was written for the Hewlett Packard 9825A Calculator, and is the baseline for the project which is now under configuration management. Any changes to this program must be accomplished by an Engineering Change Proposal.

Program Identification Data

Product Improvement Plan: 78-07-020-I

Program Title: Meteorological Data Reduction Program for the Hewlett Packard 9825A Calculator

Programming Personnel: Raymond Bellucci (formerly of the Atmospheric Science Laboratory)
Steven W. Burnett
Thomas Richter (formerly of the Atmospheric Science Laboratory)

Originating Department: *Combat Surveillance & Target Acquisition Laboratory
ERADCOM, Fort Monmouth, New Jersey 07703

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
AVAILABILITY CODES
SPECIAL
A 23 EP

*This project was started in October 1976 by the Fort Monmouth element of the Atmospheric Science Laboratory.

DDC
RECEIVED
MAR 22 1979
D

TABLE OF CONTENTS

	Page
1. PROGRAM DESCRIPTION	1
a. System Flow	3
b. Preflight Section	4
c. Flight Section	7
d. Output Section	8
e. Last Flight Section	18
f. Flight Information Section	18
g. P. Tape Section	19
h. PIBAL Section	19
i. TACFIRE Section	19
j. Tables	19
2. FILE DESCRIPTION	33
a. Track Ø	33
b. Track 1	33
c. T List Files	34
3. VARIABLE DESCRIPTION	35
a. Array Variables	35
b. String Variables	38
c. r- variables	40
4. FLOW CHARTS	41
a. System	41
b. Preflight	44
c. Flight	59
d. Output	64
e. Last Flight Update	75
f. Flight Information	76
g. P. Tape	77
h. TACFIRE	78
5. SAMPLE TEST CASE	79
a. Test Flights	80
b. Punch Output	89

TABLE OF CONTENTS

		Page
6.	PROGRAM LISTING	95
	a. System Listing	95
	b. Preflight	97
	c. Flight	113
	d. Output	120
	e. Last Flight	146
	f. Flight Information	171
	g. P. Tape	173
	h. TACFIRE	177
7.	REFERENCES	178

FIGURES

1.	Operating Instruction Flow Chart	41
2.	Autoload and Master Supervisory File (Track 0 File 0)	42
3.	Functional Keys and Program Flow	43
4.	a thru o Preflight (Track 0 Files 2 and 5)	44-58
5.	a thru e Flight (Track 0 File 6)	59-63
6.	a thru l (Output Sound Ranging, Fallout Ballistic Met 3 (Track 0 File 12)	64-75
7.	Flight Information (Track 1 File 9)	76
8.	Pressure Tape (Track 1 File 2)	77
9.	TACKFIRE (Track 1 File 3)	78

TABLES

1.	Special Function Keys	19
2.	Octant of Globe where Station is Located	20
3.	BAUDOT Code (TTY-76)	20
4.	ASCII Code (UGC-74)	21
5.	Zone Structure	23
6.	Standard Ballistic Zone Densities	24
7.	Standard Ballistic Zone Temperatures	24

TABLE OF CONTENTS

		Page
8a.	Temperature Weighting Factors (Type-3 Message) (Surface-to-Surface Trajectories)	25
8b.	Density Weighting Factors (Type-3 Message) (Surface-to-Surface Trajectories)	25
8c.	Wind Weighting Factors (Type-3 Message)	26
9a.	Temperature Weighting Factors (Type-2 Message)	27
9b.	Density Weighting Factors (Type-2 Message)	28
9c.	Wind Weighting Factors (Type-2 Message)	29
10.	NWS Format for Pressure Calibration Tape	30
11.	AN/AMT-4 Humidity Table E[14,15]	31
12.	J005 Humidity Table E[14,15]	32
13.	Humidity RD Table for AN/AMT-4 Radiosonde	37
14.	Humidity RD Table for J005 Radiosonde	38

1. PROGRAM DESCRIPTION

Purpose: The OL-192 Meteorological Data Reduction Program quickly converts raw meteorological data into accurate real time meteorological messages. Real time meteorological data produced by the OL-192 increases the artillery commander's capability for a first round hit probability. This increases the effectiveness of friendly artillery while reducing our weapons' exposure to enemy acquisition systems counter battery fires.

Use: The OL-192 is used by the artillery meteorological sections to reduce the raw meteorological data received by the Radiosonde Recorder AN/TMQ-5(*) and the winds data received by the Control Recorder C-577(*), a component of the Radiosonde System AN/GMD-1(*). Input is entered from the calculator's console for real flight time computation. The OL-192 program can also calculate messages from the last flight data, it can punch pressure calibration tapes for radiosondes, and it can print the flight information for each entered significant level and zone information for each zone height level.

Physical Description

Hewlett-Packard 9825A Calculator

Displays:

Light Emitting Diodes (LED)	- 32-character data display to prompt the operator.
Strip Printer	- Prints 16-character wide hard copy listing of programs, entries and messages.
Live keyboard	- Permits use of the computer while a program is running.
Audio Beeper	- Alerts operator for data inputs.

Remex Reader/Perforator

Tape Reader:

Tape Material :	8-channel, 1-inch wide ASCII, Mylar or paper.
Speed :	300 characters per second.
Tape Form :	Loop or strip.
Tape Direction:	Left to right.

Perforator:

Tape material:	8-channel, 1-inch wide, ASCII, Mylar or paper. 5-channel, 11/16-inch wide, BAUDOT, Mylar or paper.
Speed:	120 characters per second.
Tape form:	1,200-foot roll (2-inch core)

Reader/Perforator Rewind

Winds punch data tapes for ease of storage.

LED Display and Audible Beep Interpretation.

Types of displays and beeps

1. Verification of data entered prompt: displays data for verification and ends with a question mark (?). The YES and NO keys are activated and there is one audible beep.

2. Entry of data prompt: displays ENTER, type of data entry, and a range of entry data values. There are two audible beeps.

3. Action prompt: displays the type of action required. There are three audible beeps.

Flag Definitions (True/False):

Flag 0: Optional output

Flag 1: Figures output on BAUDOT code (TTY-76)/no figures output

Flag 2: Letters output on BAUDOT code (TTY-76)/no letters output

Flag 3: No delete/delete

Flag 4: Paper tape reader-perforator on/off

Flag 5: Figures mode on BAUDOT code (TTY-76)/no figures mode

Flag 6: Letters mode on BAUDOT code (TTY-76)/no letters mode

Flag 7: TTY-76 code (BAUDOT)

Flag 8: UGC-74 code (ASCII)

Flag 9: Continue flight/flight

Flag 10: Not used

Flag 11: Missing data/no missing data

Flag 12: AWS messages/no AWS messages.

a. System Flow

The general flow of the OL-192 program is controlled by the Master Supervisory File and the Special Function Keys. When the calculator is turned on and the program cartridge is seated, the calculator will automatically load the Master Supervisory File and begin running the program. The program prints the program title, version number and version date first. It then loads the functional key definitions from file 1 (see Table 1 for description of special function keys). The Master Supervisory File prompts the operator to press one of the upper row keys that activate one of six other routines. These routines are:

1. START - Start flight procedures.
2. CONT - Continue current flight from last entered data level or zone.
3. LAST FLT - Print/Punch meteorological messages from the last flight data.
4. P. TAPE - Prompt operator through sequence to punch NWS format radiosonde pressure tape.
5. PIBAL - Pilot balloon method of data entry. Not implemented.
6. TACFIRE - TACFIRE test-loop sequence to test proposed tacfire interface.

The other six functional keys are activated by the individual routines. They are:

1. INFO - Print flight level and zone level information for the flight record.
2. OUTPUT - Produce meteorological messages from light level and zone level data.
3. DELETE - Delete last significant data level, message or contact.
4. NO - No answer key.
5. YES - Yes answer key.
6. DATA IN - Used to input data into calculator.

An operator's flow chart and a system flow chart showing the interconnection of the routines is in section 4.a (P. 41).

The OL-192 Meteorological Real Time Program is divided into three logical sections.

1. Preflight (surface data and baseline check).

2. Flight (inflight data reduction).

3. Output (meteorological messages).

The preflight section is entered when the operator presses the "START" key. After the preflight section, the program enters the flight section. At any time in the flight section, the operator can enter a NO key answer and activate the INFO, OUTPUT, and DELETE keys. The operator can delete significant levels, print flight information, or get meteorological messages by pressing one of these keys. The Output section is entered by pressing the "OUTPUT" key. After any of the three actions is completed, the program will return to the next significant level or zone level entry.

b. Preflight Section

The preflight section is entered by pressing the "START" key. The preflight routine prompts the operator through entry of the Date-Time Group, surveys the site of launch, enters the type of output, checks the recorder linearity, enters the radiosonde type, and enters the pressure table by the 15-contact or 180-contact method. It also enters the temperature and humidity tables, the baseline data values and the meteorological data at launch.

(1) Date-Time Group

The DATE-TIME Group routine prompts the operator to enter the year, month, day and time in GMT (ZULU) and assembles the DATE-TIME Group. The group is used to update the flight date, survey date, linearity date and last flight date. The DATE-TIME Group is stored in D$.

(2) Survey Data

If the previous flight launch site is the same as the current flight, the previous survey data can be used. These data are stored in P$ and are printed for the flight record. Otherwise, the routine prompts the operator through entry of the station elevation. If the launch site is a coded location, the code is entered; otherwise the latitude, longitude and quadrant (see Table 2) are entered. The routine prints the survey data for verification and flight record. It updates P$ and the flight launch site date O$. In the messages, the headers contain the site identification.

(3) Output Type

At this time, the operator decides what type of punched teletape will be punched by the OUTPUT section. It can be either TTY-76 (BAUDOT code - 5 channel - 11/16-inch tape: see Table 3) or UGC-74 (ASCII code - 8-channel - 1-inch tape: see Table 4). The ASCII code is used internally by the calculator. When punching BAUDOT tape, the OUTPUT section converts the ASCII code to BAUDOT using a string array of 64 characters. The character is looked up in the array H$. The position converts into a number between 0 and 31. This number is punched in binary on a five channel tape. The figures and letters mode is determined by which 32 character subarray the character is found in. If the character is not in the respective mode, the mode key (figures or letters) is punched before punching the character.

(4) Recorder Linearity

If the previous linearity date L$ is still valid and the recorder serial number N$ is the same, the previous linearity data are printed for the flight record. If invalid, the operator enters the recorder serial number and recorder calibration values at 190, 180, 160, 140, 120, 100, 80, 60, 40, 20, and 10 cycles. These values are stored in G$ and are used to correct recorder readings in the subroutine CORRECT READING. The program rejects a recorder where any printed trace exceeds ± 0.3 of a recorder division. The routine prints a recorder linearity table, listing the deviations between the true and actual values for the flight record.

(5) Radiosonde Type

This routine enters the type of radiosonde used in the flight. If it is an AN/AMT-4, the program will calculate the temperature and humidity using the AN/AMT-4 humidity RD (array K) and humidity (array E) tables. If it is a JOO5 radiosonde, the JOO5 humidity RD and humidity tables are loaded. The program prints after the entry of the baseline values, the baseline ratios for the lock-in of the Computer, Humidity-Temperature CP-223C/UM if the radiosonde is an AN/AMT-4. If it is a JOO5 radiosonde the program will print the baseline temperature ratio setting for the USWB-230 computer and the humidity ratio setting for the USWB-300 computer.

(6) Pressure Table

There are two ways to enter the radiosonde pressure values for each contact.

15-contact method: This method inputs 15 selected contact values. The program checks that these 15 values form a pressure curve that is proportional to the standard curve U$ in the calculator. The possible deviation from the curve is ± 6 mb at high pressure down to ± 1 mb at low pressure.

180-contact method: This method inputs all 180 contact values. The 15-contact values are selected from this table.

Therefore, the method of computing pressures is the same whether the 15- or 180-contact entry method is used. The program uses the given 15 contacts or, in the 180-contact method, extracts the 15 selected contacts in the 15-contact array V$. Linear interpolation is used to calculate the pressure for a given contact and the contact for a given pressure. The pressure is calculated using the 15-contact values and the curve generated by the standard pressure table or by the 180-contact pressure table.

If the operator has a pressure tape, the pressure tape is loaded into the calculator, the 15-contact values are extracted, and the program proceeds to check the pressure calibration table. If the operator does not have a pressure tape, the program enters the 15-contact values. The program checks the interpolated midpoints against the standard table interpolated midpoints. If the values are bad ($\pm$6 mb), the program loads the pressure tape making routine P. TAPE. If the REMEX is down, the program allows the operator to enter manually the 180 contacts into the calculator. Otherwise, the program prompts the operator through making a pressure tape and restarts the flight.

(7) Baseline Data

This routine enters the wet and dry bulb temperatures, recorder reference reading, temperature trace reading and humidity trace reading from the baseline chamber. If the entered values are verified, the program will print out the values after correcting them for drift and frequency. If the humidity is less than 20 percent or greater than 70 percent or the temperature is less than 20°C or greater than 30°C, the baseline check is repeated. The values entered are stored in array B(*).

(8) Launch Data

This routine calculates the geopotential height constant, enters the azimuth offset, horizontal distance offset, surface wind direction, surface wind speed, and surface pressure and contact. These values are verified and stored in array A(*). In the flight section, they will be stored in flight array F(*).

(9) Contact Correction

If the surface contact has changed between the launch data input and the pressure table verification, this routine will calculate an automatic pressure correction. This value is printed for the flight record. If the contact corrections are fewer than $\pm$2.0 contacts, the program automatically applies the correction to the significant level contact pressure values. If the automatic contact corrections are more than 2.0 contacts, the flight is terminated.

(10) Air Weather Service Message Flag:

This routine enters the operator's desire for the Air Weather Service (AWS) messages in the OUTPUT and last flight sections. If the messages are needed, this routine enters the USAF location code for the message header lines.

The program now loads the flight section of the program.

c. Flight Section.

This section enters all flight data for each significant level and each zone height level. It is initiated by entry from the preflight section,or the current flight can be continued by pressing the "CONT" key. The significant level data is entered for each level passed. If the programs determines that a zone height has been passed, it calculates the time the zone level data are needed and inputs the data.

(1) Significant Levels

This routine enters the contact, time, temperature and humidity for each significant level and prints these data for verification. If verified, the program calculates and displays the temperature lapse rate. The significant level data are stored in the flight table array F(*). The missing data routine is entered when a star (*) is entered with the reference value. The routine will print "MISSING DATA START HERE." The last good data level is printed using INFO and the next good data level is entered and printed using INFO. The entry of significant level data is continued. The missing mandatory levels are saved in array F(*) and printed as missing levels. The maximum data level is 60.

(2) Zone Level Data

This routine enters azimuth and elevation angles for a tenth of a minute below and above the zone height. The angles are printed and, if verified, the message height is displayed. The program interpolates for the exact angles. The recorder increments for time can be set by the operator. When the zone level elevation is requested, an asterisk (*) is entered with the incremented value. Thereafter, the program will request angular data with that time increment. The time increment values are 1.0, or 0.5, and the standard incremented value is 0.1 minutes. Missing angles can be interpolated by entering the elevation angle with a star (*). The program requests the time the missing angles start and the time the missing angles stop. It then requests the elevation and azimuth angles at these two points. The program then interpolates the angles for the zones between the two times. After the balloon has reached 30 kilometers, there is an automatic output of meteorological messages. The maximum zone is 45 which is equivalent to 55 kilometers. Zone level data are stored in the array H (*).

d. OUTPUT Section

This section is entered from the Flight section. In the flight section, a NO answer to any data input query will activate the OUTPUT key. The Output program produces meteorological messages from the flight data in the flight level array F(*) and the zone level array H(*) tables that were entered in the Flight section. The OUTPUT section computes and outputs the following messages from the current flight:

SOUND RANGING
COMPUTER MET (FADAC M-18)
FALLOUT
BALLISTIC 3
BALLISTIC 2
AWS MESSAGES
TTAA
TTBB
PPBB
TTCC
TTDD
PPDD

After print/punching the messages, the last flight output may be updated by the current flight. The last flight messages section is exactly the same as the output section except that last flight uses data on track one.

(1) Sound Ranging Message

The general requirement is to determine the windspeed and direction in four layers from the surface to 800 meters and to determine the virtual temperature at a height of 200 meters. The data transmitted are the effective temperature, effective wind direction, effective windspeed and the time of release. The effective temperature is the sonic temperature at 200 meters. For the effective wind direction and speed, the sound ranging layer wind calculations are the weighted and summed X, Y wind components for the surface, 200, 400, 600 and 800 meter levels:

(a) The Weighting Factors are:

Structure	Height	Surface	200	400	600	800
Normal		0.2	0.5	0.15	0.075	0.075
2		0.4	0	0.3	0.15	0.15
3		0	1.0	0	0	0
4		0	0	1.0	0	0

Normal Structure - The 400-meter layer wind is one to two times the 200-meter layer wind.

Structure 2 - The 400-meter layer wind is greater than two times the 200-meter layer wind.

Structure 3 - The 400-meter layer wind is less than the 200-meter layer wind and within two knots of the surface wind.

Structure 4 - The 400-meter layer wind is less than the 200-meter layer wind and not within two knots of surface wind.

(b) Sound Ranging Message Format

METSRQXXXXXX
YYGGGV±TTTDDDFF
9

METSR	Identifying prefix for sound ranging message
Q	Octant of the globe (0 to 8 not 4), 9 if location is coded.
XXXXXX	Location of reporting station. The first three digits encode latitude and the last three encode longitude in degrees and tenths of degrees.
YY	Day of month (01 to 31)
GGG	Hour of valid period (GMT) in tens, units and tenths of hours (000 to 239).
V	Length of valid period in hours (1 to 8), 9 equals 12 hours. (Set to Ø)
±TTT	Effective temperature to nearest 1/10th degree C (-499 to +499).
DDD	Effective wind direction in tens of mils (001 to 640). (000 for calm wind).
FF	Effective wind speed in knots (00 to 99)
9	Message terminator. (Punch output)

(2) Computer Meteorological Message (FADAC M-18)

The computer message differs from the ballistic (NATO) message in that the zoning structure is different, the zone values are not

weighted, and the weather elements are reported as true values instead of weighted percent of standard. The elements reported for each of the 26 zones (see Table 5) are wind direction, wind speed, virtual temperature, and the zone midpoint pressure.

Computer Meteorological Message Format

METCMQXXXXXX
YYGGGVHHHPPP
ZZDDDFFFTTTTRRRR
xxxxxxxxxxxxxxxx
26DDDFFFTTTTRRRR
9

METCM	Identifying prefix for computer message.
Q	Octant of the globe
XXXXXX	Location of reporting station. The first three digits encode latitude and the last three encode longitude in degrees and tenths of degrees.
YY	Day of month (01 to 31)
GGG	Hour of valid period (GMT) in tens, units and tenths (000 to 239)
V	Length of valid period hours (1 to 8), 9 equals 12 hours. (Set to Ø)
HHH	Altitude of meteorological datum plane in tens of meters above mean sea level.
PPP	Station pressure to nearest millibar (omit thousands digit)
ZZ	Line number for message (00 to 26)
DDD	Zone wind direction in tens of mils (001 to 640) (000 for calm wind)
FFF	Zone wind speed in knots (000 to 999)
TTTT	Zone virtual temperature to nearest 0.1 degree K (0000 to 5000)
RRRR	Zone midpoint pressure in millibars (0000 to 1100)
9	Message terminator (Punch output)
/	Missing data indicator

(3) FALLOUT Message

The Meteorological Message for fallout contains the average vector wind for each 2,000 meter zone from the surface to a height of 30,000 meters (see Table 5). The average wind for each 2,000-meter zone is reported to the nearest one knot and to the nearest 10 mils. The distance traveled in each fallout zone is calculated from the zone level data and an average speed and direction are computed. Fallout winds are not weighted:

Fallout Message Format

METFMQXXXXXX
YYGGGVHHHTRO
ZZDDDFFF
xxxxxxxx
15DDDFFF
9

METFM	Identifying prefix for fallout message.
Q	Octant of the globe (0 to 8 not 4), 9 if location is coded.
XXXXXX	Location of reporting station. The first three digits encode latitude and the last three encode longitude in degrees and tenths of degrees.
YY	Day of month (01 to 31).
GGG	Hour of valid period (GMT) tens, units and tenths (000 to 239).
V	Length of valid period (1 to 8), 9 equals 12 hours (Set to Ø).
HHH	Altitude of meteorological datum plane in tens of meters above mean sea level (000 to 999).
ZZ	Line number for fallout data (00 to 15).
DDD	Wind direction to nearest 10 mils (001 to 640) (000 for calm winds).
FFF	Wind speed in knots (000 to 999).
9	Message terminator (Punch output).
/	Missing data indicator.
TRO	Filler for fallout message.

(4) Ballistic Meteorological Messages

The ballistic message is a measure of the parameters of the atmosphere, a comparison of the current conditions with standard conditions, and a report of the variations in terms of weighted percents of standard (see Tables 6 and 7). The average wind speed and direction for each of the atmospheric zones are determined. The zone values of density and temperature are compared with the standard zone values and variations from the standard are determined. The variations from standard are then weighted according to specified zone weighting factors. These mean weighted quantities are the ballistic values.

Two general categories of trajectoris have been established:

Type 3 - surface to surface (see Table 8 for weights),

Type 2 - surface to air (see Table 9 for weights).

Ballistic Message Format:

```
METBKQXXXXXX
YYGGGVHHHPPP
ZZDDFFTTTRRR
XXXXXXXXXXXX
15DDFFTTTRRR
9
```

METB	Identifying prefix for ballistic messages.
K	Type of message (the value of K may be 2 for ballistic met 2, or 3 for ballistic met 3).
Q	Octant of the globe (0 to 8 not 4), 9 when station is coded.
XXXXXX	Location of reporting station.
YY	Day of the month (01 to 31).
GGG	Hour of beginning valid period (GMT) tens, units, and tenths (000 to 239).
V	Length of valid period hours (1 to 8), 9 equals 12 hours (Set to Ø).
HHH	Altitude of meteorological datum plane in tens of meters above mean sea level.
PPP	Station pressure in percent of ICAO standard to nearest 0.1 percent (000 to 999).
ZZ	Line number for ballistic information (00 to 15).

DD	Ballistic wind direction in hundreds of mils (01 to 64) (00 for calm wind).
FF	Ballistic wind speed in knots (00 to 99). When wind speed equals or exceeds 100 knots, add 80 to the line number.
TTT	Ballistic temperature in percent of ICAO standard to nearest 0.1 percent (000 to 999), drop hundreds digit.
RRR	Ballistic density in percent of ICAO standard to the nearest 0.1 percent (drop hundreds digit).
9	Message terminator (Punch output).
/	Missing data indicator.

(5) Air Weather Service Message

The AWS meteorological message is comprised of a single meteorological bulletin preceded by a header and followed by the end of transmission signal(;). The message is composed of figure groups, each group has significance according to its position in the message following the section indicator or a particular self-identifying group. When observed datum is not available for an element, the appropriate code figure or missing indicator (/) is reported.

The AWS message is divided into four parts (A, B, C and D) for coding purposes. Only data at or below 100 mb are reported in PARTS A and B. Only data above 100 mb are reported in PARTS C and D.

The submessages reported are:

AWS First Transmission

TTAA (PPAA)	- data up to and including 100 mb at isobaric surfaces.
TTBB	- data up to and including 100 mb. Significant levels with respect to temperature and/or humidity.
PPBB	- Wind data for fixed regional heights up to and including 100 mbs.

AWS Second Transmission

TTCC (PPCC)	- Data above 100 mbs at isobaric surfaces.
TTDD	- Data above 100 mbs to termination. Significant levels with respect to temperature and/or humidity.
PPDD	- Wind data for fixed regional heights above 100 mbs.

(a) Levels Reported

The mandatory levels reported in PARTS A and C are the standard isobaric surfaces of 1000, 850, 700, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10, 8, 5, 3, 2, and 1 mb. When the geopotential height of one (or more) standard isobaric surface is lower than the altitude of the station, the level is reported and the height is trended down. Solidi are reported for temperature, dew point depression and the wind group.

Significant levels are reported in PARTS B and D. They include:

Surface data
Highest and lowest temperature (nearest 0.1°C)
Highest and lowest humidity
Termination level for humidity
Termination level of the flight

A sufficient number of levels are included so that for pressures of 300 mb or greater, the temperature at any level between two adjacent significant levels selected for coding should not differ more than 1°C from that obtained by linear interpolation between the two selected levels. At pressures lower than 300 mb, the difference should not exceed 2°C. For the humidity criteria, the relative humidity between any two adjacent significant levels does not differ by more than 10 percent from that obtained by linear interpolation between levels.

The tropopause level is defined as the lowest level at which the lapse rate decreases to 2°C/km or less, and the average lapse rate from this level to any level within the next higher 2 km does not exceed 2°C/km. If the ascent does not reach at least 200 mb, no attempt is made to determine a tropopause. If the ascent terminates less than 2 km above a level that appears to be a tropopause, a tropopause is not determined. If no tropopause level is found, a 88999 code group is reported.

The maximum wind level is defined as the level where the wind speed is greater than that observed immediately above and below that level. It must be above the 500 mb level and it must have a speed greater than 60 knots. It is determined from the zone height levels. A 77999 code group is reported for a missing maximum wind if the ascent reaches 500 mb.

(b) Air Weather Service Message Format

```
METWQAAAGGOOOgg
YYhhh
TTAA
;
TTBB
;
PPBB
;
TTCC
;
TTDD
;
PPDD
;
```

METW	Designates Air Weather Service Meteorological Message.
Q	Octant of the globe (0 to 8 not 4), 9 if location is coded.
AAA	Latitude in degrees and tenths (000 to 900).
GG	Hour of balloon release (00 to 23).
OOO	Longitude in degrees and tenths (000 to 999). When longitude is equal to or greater than 100 degrees, the hundreds digit is dropped.
gg	Minutes of balloon release (01 to 59).
YY	Day of the month (01 to 31).
hhh	Altitude of the station above mean sea level in tens of degrees.

First Transmission - (data up to and including 100 mb level)

Part A - Standard Isobaric Surfaces

ØTTAAYYGGIXXXXX		Identification
99PPPTTTDDddfff		Surface Data
00hhhTTTDDddfff		
85		Standard Isobaric Surfaces
70		
50		
40		
30		
25		
20		
15		
10		
88PPPTTTDDddfff	or 88999	Tropopause Data
77PPPddfff4bbaa	or 77999	Maximum Wind data
;		

Part B - Significant Levels with respect to temperature and/or humidity

ØTTBBYYGG/XXXXX	Identification
00PPPTTTDD	Surface Data
IIPPPTTTDD	
22	Significant Levels
33	
44	
etc	
;	

Fixed Regional Wind Levels

ØPPBBYYGGØXXXXX	Identification
90012	0, 1, 2-thousand foot wind
ddfffddfffddfff	
90346	3, 4, 6
ddfffddfffddfff	
90789	7, 8, 9
ddfffddfffddfff	
91246	12, 14, 16
ddfffddfffddfff	
9205/	20, 25
ddfffddfffddfff	
9305/	30, 35
ddfffddfff	
950X/	50,000-foot wind and 100 mb level
ddfffddfff	
;	

Second Transmission - (Data above 100 mb)

Part C - Standard Isobaric Surfaces

ØTTCCYYGGIXXXXX	Identification
70hhhTTTDDddfff	
50	Standard Isobaric
30	
20	
10	
07	
05	
03	
02	
01	
88PPPTTTDDddfff or 88999	Tropopause Data
77PPPddfff4bbaa or 77999	Maximum wind data
;	

Part D - Significant Levels with Respect to Temperature and/or Humidity

ØTTDDTTGG/XXXXX	Identification
IIPPPTTTDD	
22	Significant levels
33	
44	
etc	
;	

Fixed Regional Wind Levels

ØPPDDYYGGØXXXXX	Identification
970//	70,000-foot wind
ddfff	
990//	90
ddfff	
110//	110
ddfff	
140//	140
ddfff	170,000-foot wind and
170X/	termination of flight level
ddfffddfff	
;	

(c) Definition of AWS Symbols

Ø Blank space

DD Depression of the dew point temperature (with respect to water) at the specified standard isobaric surface or significant level. When the depression is 4.9°C or less, it is reported in tenths of a degree. When the depression is more than 4.9°C, it is reported in whole degrees plus 50.

dd True direction in tens of degrees (the hundreds and tens digits of the observed direction rounded off to the nearest 5°) from which the wind is blowing at the specified level.

fff Wind speed in knots, or knots plus 500, if the wind direction ends in 5°.

GG Actual time of observation to the nearest whole hour in GMT.

hhh Geopotential height in whole meters below 500 mb, and tens of meters for 500 mb and above, for the standard isobaric surfaces.

I Wind indicator that specifies the hundreds digit (Part A) and the tens digit (Part C) of the pressure of the last standard isobaric surface for which a wind group is reported.

XXXXX USAF Position Location Code

PPP Pressure at the significant levels. In Part B (up to and including 100 mb), these pressures are reported in whole millibars. In Part D (above 100 mb), pressures are reported in tenths of a millibar.

TTT Approximate temperature value in tenths of degree Celsius. If the tenths digit is even, the temperature is positive. If the tenths digit is odd, the temperature is negative.

bb Wind shear between the maximum wind and a level 1000 meters below.

aa Wind shear between the maximum wind and a level 1000 meters above.

YY Day of the month plus 50 to indicate that the wind speeds are in knots.

e. Last Flight Section

This section calculates and prints the meteorological messages for the significant level data table and the zone level data table of the last flight that the operator used to update these files. This routine is also used by the operator as an operational check of the calculator. This section is entered by pressing the "LAST FLT" key. The last flight routine is exactly the same as the output section except that the last flight uses data files on track one instead of track zero.

f. Flight Information Section

This section is entered by pressing the "INFO" key. This routine prints out significant level and zone level data. It is useful in three instances:

1. To know the actual meteorological values at any levels desired.

2. To determine the pressures at the bottom and top of layers of missing data when applying missing data rules.

3. To print after flight termination all processed data for mission record filing purposes.

g. P. TAPE Section

This section is entered by pressing the "P. TAPE" key. This program prompts the operator through the punching of a 180-contact pressure tape in the NWS format (see Table 10 for example). This tape is used in the preflight section to enter the baroswitch calibration chart.

h. PIBAL Section

Not implemented.

i. TACFIRE Section

This routine prints and displays a test loop for a proposed tacfire interface. The interface is to be connected to the REMEX punch circuit for automatic sending when the output is punched on the REMEX Reader/Perforator.

j. Tables:

TABLE 1 - SPECIAL FUNCTION KEYS

F_0	START	-	Starts meteorological data processing in preflight section.
F_1	CONT	-	Continues meteorological processing in flight section.
F_2	LAST FLT	-	Calculates meteorological messages for last flight.
F_3	TACFIRE	-	TACFIRE system test-loop.
F_4	P. TAPE	-	Programmed to prompt operator through procedures to punch radiosonde pressure calibration tape.
F_5	PIBAL	-	Not implemented.
F_6	INFO	-	Activates after operator presses NO or STOP key. When pressed, the program prints processed meteorological data beginning with the last significant level down to the surface, pauses, then prints processed wind data from surface up to the height of the last meteorological zone level.
F_7	OUTPUT	-	Activates after operator presses NO or STOP key. When pressed, LED display prompts the operator to turn on the tape perforator, if punched copy of all meteorological messages is desired. The computer outputs printed hard copies of all meteorological messages.

F_8 - DELETE - Activates after operator presses NO or STOP key. Deletes punch/print of all unwanted meteorological messages. Repeating STOP - DELETE allows back stepping to correct errors or pick up omitted data for processing.

F_9 - NO - Used as verifier key, causes repeat of data input sequence.

F_{10} - YES - Used as verifier key, entered data is saved, and printed for the flight record.

F_{11} - DATAIN - Enters data into calculator.

TABLE 2 - OCTANT OF GLOBE WHERE STATION IS LOCATED

0	0° - 90°	West longitude	North latitude
1	90°- 180°	"	"
2	180°- 90°	East longitude	"
3	90°- 0°	"	"
4	Not used		
5	0° - 90°	West Longitude	South Latitude
6	90°- 180°	"	"
7	180° - 90°	East Longtitude	"
8	90°- 0°	"	"
9	Coded location indicator		

TABLE 3 - BAUDOT CODE (TTY-76)

Binary	Letters	Figures	Binary	Letters	Figures
0	Null	Null	16	E	3
1	T	5	17	Z	"
2	CR	CR	18	D	$
3	0	9	19	B	?
4	Space	Space	20	S	Bell
5	H	#	21	Y	6
6	N	,	22	F	!
7	M	.	23	X	/
8	LF	LF	24	A	-
9	L	)	25	W	2
10	R	4	26	J	'
11	G	&	27	FI	FI
12	I	8	28	U	7
13	P	Ø	29	Q	1
14	C	:	30	K	(
15	V	;	31	LE	LE

CR - Carriage return; LF - Line feed; FI - Figures; LE - Letters.

TABLE 4 - ASCII CODE (UGC-74)

ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec	ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec
NULL	00000000	000	0	DLE	00010000	020	16
SOH	00000001	001	1	DC_1	00010001	021	17
STX	00000010	002	2	DC_2	00010010	022	18
ETX	00000011	003	3	DC_3	00010011	023	19
EOT	00000100	004	4	DC_4	00010100	024	20
ENQ	00000101	005	5	NAK	00010101	025	21
ACK	00000110	006	6	SYNC	00010110	026	22
BELL	00000111	007	7	ETB	00010111	027	23
BS	00001000	010	8	CAN	00011000	030	24
HT	00001001	011	9	EM	00011001	031	25
LF	00001010	012	10	SUB	00011010	032	26
Vtab	00001011	013	11	ESC	00011011	033	27
FF	00001100	014	12	FS	00011100	034	28
CR	00001101	015	13	GS	00011101	035	29
SO	00001110	016	14	RS	00011110	036	30
SI	00001111	017	15	US	00011111	037	31
space	00100000	040	32	N	01001110	116	78
!	00100001	041	33	O	01001111	117	79
"	00100010	042	34	P	01010000	120	80
#	00100011	043	35	Q	01010001	121	81
$	00100100	044	36	R	01010010	122	82
%	00100101	045	37	S	01010011	123	83
&	00100110	046	38	T	01010100	124	84
'	00100111	047	39	U	01010101	125	85
(	00101000	050	40	V	01010110	126	86
)	00101001	051	41	W	01010111	127	87
.	00101010	052	42	X	01011000	130	88
+	00101011	053	43	Y	01011001	131	89
,	00101100	054	44	Z	01011010	132	90
-	00101101	055	45	[	01011011	133	91
.	00101110	056	46	\	01011100	134	92
/	00101111	057	47	]	01011101	135	93
Ø	00110000	060	48	^	01011110	136	94
1	00110001	061	49	_	01011111	137	95
2	00110010	062	50	`	01100000	140	96
3	00110011	063	51	a	01100001	141	97
4	00110100	064	52	b	01100010	142	98
5	00110101	065	53	c	01100011	143	99
6	00110110	066	54	d	01100100	144	100
7	00110111	067	55	e	01100101	145	101
8	00111000	070	56	f	01100110	146	102
9	00111001	071	57	g	01100111	147	103
:	00111010	072	58	h	01101000	150	104
;	00111011	073	59	i	01101001	151	105
<	00111100	074	60	j	01101010	152	106
=	00111101	075	61	k	01101011	153	107
>	00111110	076	62	l	01101100	154	108
?	00111111	077	63	m	01101101	155	109
@	01000000	100	64	n	01101110	156	110

ASCII Char.	EQUIVALENT FORMS Binary	Octal	Dec
A	01000001	101	65
B	01000010	102	66
C	01000011	103	67
D	01000100	104	68
E	01000101	105	69
F	01000110	106	70
G	01000111	107	71
H	01001000	110	72
I	01001001	111	73
J	01001010	112	74
K	01001011	113	75
L	01001100	114	76
M	01001101	115	77

ASCII Char.	Binary	Octal	Dec
o	01101111	157	111
p	01110000	160	112
q	01110001	161	113
r	01110010	162	114
s	01110011	163	115
t	01110100	164	116
u	01110101	165	117
v	01110110	166	118
w	01110111	167	119
x	01111000	170	120
y	01111001	171	121
z	01111010	172	122
{	01111011	173	123
|	01111100	174	124
}	01111101	175	125
~	01111110	176	126
DEL	01111111	177	127

TABLE 5 - ZONE STRUCTURE.

Zone structure of the NATO, computer, and fallout metro messages

Height Meters	Line numbers		
	NATO	Computer	Fallout
Surface	0	0	0
200	1	1	1
500	2	2	
1000	3	3	
1500	4	4	
2000	5	5	
2500	6	6	2
3000		7	
3500	7	8	
4000		9	
4500	8	10	3
5000		11	
6000	9	12	
7000	10	13	4
8000		14	
9000	11	15	5
10000		16	
11000	12	17	6
12000		18	
13000	13	19	7
14000		20	
15000	14	21	8
16000		22	
17000	15	23	9
18000		24	
19000		25	10
20000		26	
* * *			* * *
30000			15

TABLE 6 - STANDARD BALLISTIC ZONE DENSITIES

Zone No.		
	0	1225.0
	1	1213.3
	2	1184.4
	3	1139.2
	4	1084.6
	5	1032.0
	6	957.0
	7	863.4
	8	777.0
	9	697.4
	10	590.0
	11	467.0
	12	364.8
	13	266.6
	14	194.8
	15	142.3

TABLE 7 - STANDARD BALLISTIC ZONE TEMPERATURES

Zone No.		
	0	288.2
	1	287.5
	2	285.9
	3	283.3
	4	280.0
	5	276.8
	6	271.9
	7	265.5
	8	259.0
	9	252.5
	10	242.7
	11	229.8
	12	216.8
	13	216.7
	14	216.7
	15	216.7

TABLE 8. a. TEMPERATURE WEIGHTING FACTORS (TYPE-3 MESSAGE)
(SURFACE-TO-SURFACE TRAJECTORIES)

T3

Line Number	Zone No. 1	2	3	4	5	6	7	8	9	10-15
1	1.00									
2	.27	.73								
3	.13	.20	.67							
4	.08	.12	.25	.55						
5	.05	.10	.20	.21	.44					
6	.04	.04	.09	.11	.13	.59				
7	.02	.04	.07	.09	.11	.26	.41			
8	.01	.03	.05	.04	.10	.19	.23	.35		
9-15	.01	.01	.02	.03	.03	.09	.13	.24	.44	.00

b. DENSITY WEIGHTING FACTORS (TYPE-3 MESSAGE)
(SURFACE-TO-SURFACE TRAJECTORIES)

D3

Line Number	Zone No 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.43	.57													
3	.22	.31	.47												
4	.15	.21	.32	.32											
5	.11	.17	.25	.22	.25										
6	.08	.11	.17	.17	.15	.32									
7	.06	.08	.14	.13	.12	.22	.25								
8	.05	.06	.11	.11	.10	.19	.17	.21							
9	.04	.06	.09	.09	.08	.17	.15	.14	.18						
10	.03	.04	.07	.07	.07	.13	.12	.11	.11	.25					
11	.01	.03	.05	.05	.06	.12	.11	.09	.09	.16	.23				
12	.02	.03	.05	.05	.05	.11	.10	.09	.08	.14	.12	.16			
13	.02	.02	.04	.05	.05	.11	.09	.09	.08	.14	.10	.09	.12		
14	.02	.03	.05	.05	.05	.10	.09	.08	.07	.13	.11	.08	.06	.08	
15	.02	.04	.05	.05	.05	.10	.09	.08	.07	.12	.09	.08	.05	.05	.06

TABLE 8 (Cont)

c. WIND WEIGHTING FACTORS (TYPE-3 MESSAGE)

WW

Line No.	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.20	.80													
3	.09	.19	.72												
4	.06	.12	.26	.56											
5	.04	.08	.15	.20	.53										
6	.03	.05	.08	.09	.12	.63									
7	.02	.03	.07	.07	.08	.20	.53								
8	.02	.02	.06	.06	.06	.14	.19	.45							
9	.02	.02	.05	.05	.05	.12	.13	.20	.36						
10	.01	.02	.02	.04	.03	.07	.08	.09	.09	.55					
11	.00	.00	.01	.04	.03	.08	.08	.09	.09	.20	.38				
12	.00	.01	.01	.02	.04	.07	.07	.07	.08	.17	.16	.30			
13	.00	.01	.01	.01	.03	.07	.07	.07	.07	.15	.14	.13	.24		
14	.00	.01	.01	.01	.02	.07	.07	.07	.07	.13	.13	.13	.10	.18	
15	.00	.01	.01	.01	.02	.07	.07	.07	.07	.12	.12	.11	.10	.08	.14

TABLE 9. a. TEMPERATURE WEIGHTING FACTORS (TYPE-2 MESSAGE)

T2

Line No.	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13-15
1	1.00												
2	.63	.37											
3	.37	.37	.26										
4	.25	.30	.35	.10									
5	.20	.24	.30	.18	.08								
6	.13	.19	.24	.18	.14	.12							
7	.10	.14	.20	.16	.14	.19	.07						
8	.09	.10	.17	.15	.13	.20	.12	.04					
9	.07	.09	.14	.13	.12	.19	.15	.08	.03				
10	.05	.08	.12	.10	.10	.17	.14	.10	.08	.06			
11	.05	.06	.10	.09	.08	.15	.13	.12	.10	.12	.00		
12	.04	.06	.10	.08	.08	.14	.13	.11	.10	.16	.00	.00	
13-15	.05	.06	.10	.09	.08	.16	.12	.13	.11	.10	.00	.00	.00

TABLE 9 (Cont)

b. DENSITY WEIGHTING FACTORS (TYPE-2 MESSAGE) (SURFACE-TO-AIR TRAJECTORIES)

D2

Line No.	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.63	.37													
3	.37	.37	.26												
4	.25	.30	.35	.10											
5	.20	.24	.30	.18	.08										
6	.13	.19	.24	.18	.14	.12									
7	.10	.14	.20	.16	.14	.19	.07								
8	.09	.10	.17	.15	.13	.20	.12	.04							
9	.07	.09	.14	.13	.12	.19	.15	.08	.03						
10	.05	.08	.12	.10	.10	.17	.14	.10	.08	.06					
11	.04	.06	.10	.08	.08	.15	.13	.10	.10	.12	.04				
12	.04	.06	.09	.08	.08	.13	.12	.10	.08	.13	.07	.02			
13	.03	.05	.08	.08	.06	.12	.11	.10	.08	.13	.09	.05	.02		
14	.03	.05	.06	.07	.07	.11	.10	.09	.08	.13	.10	.06	.04	.01	
15	.02	.05	.06	.07	.05	.11	.10	.08	.08	.13	.10	.07	.05	.03	.00

TABLE 9 (Cont)

c. WIND WEIGHTING FACTORS (TYPE-2 MESSAGE)

WW

Line No.	Zone No. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00														
2	.50	.50													
3	.29	.33	.38												
4	.18	.23	.39	.20											
5	.13	.18	.31	.27	.11										
6	.08	.12	.22	.20	.19	.19									
7	.07	.08	.16	.15	.16	.27	.11								
8	.04	.08	.13	.12	.13	.24	.18	.08							
9	.04	.06	.11	.10	.10	.21	.20	.12	.06						
10	.03	.04	.08	.08	.08	.16	.15	.14	.13	.11					
11	.02	.04	.06	.07	.06	.13	.13	.12	.11	.18	.08				
12	.03	.04	.07	.07	.07	.12	.11	.10	.08	.15	.10	.06			
13	.02	.04	.05	.06	.06	.11	.10	.09	.08	.14	.11	.09	.05		
14	.02	.04	.05	.06	.04	.09	.09	.09	.08	.13	.11	.09	.06	.05	
15	.01	.03	.05	.04	.05	.09	.09	.08	.07	.12	.10	.09	.08	.06	.04

TABLE 10 - NWS FORMAT FOR PRESSURE CALIBRATION TAPE

		*	*1	*2						
99999	111	1111	50	0 1272						
000		1067.3	1055.3	1038.8	1027.8	1016.8	1000.3	988.8	977.3	965.8
010	954.8	940.3	928.8	917.3	906.3	895.8	882.3	871.3	860.8	849.8
020	839.8	826.8	816.3	806.3	796.3	786.3	773.3	763.8	753.8	743.8
030	734.3	723.8	715.3	705.3	695.8	686.8	675.8	666.3	656.8	648.3
040	639.3	628.3	619.8	610.8	602.3	593.8	585.3	576.8	568.8	560.8
050	552.8	542.8	534.3	526.8	518.8	511.3	501.8	494.3	486.8	479.3
060	472.3	464.3	457.3	450.3	443.3	436.3	427.8	420.8	413.8	407.3
070	400.8	382.3	385.8	379.3	372.8	366.8	360.8	354.3	348.3	342.3
080	336.3	328.8	322.8	317.3	311.3	305.8	299.3	293.8	288.3	282.8
090	277.3	272.3	267.3	261.8	256.8	252.3	245.8	240.8	236.3	231.3
100	226.8	220.8	216.8	212.8	207.3	203.3	198.8	194.8	190.3	186.3
110	182.3	177.3	173.3	169.3	165.3	161.8	156.8	153.3	149.8	146.3
120	142.8	138.8	135.3	132.3	129.3	125.8	121.8	118.8	115.8	112.8
130	109.8	106.3	103.3	100.3	97.8	94.8	91.8	88.3	85.3	82.3
140	79.3	78.3	77.3	76.3	75.3	74.3	73.3	72.3	71.3	70.3
150	69.3	68.3	67.3	66.3	65.3	64.3	63.3	62.3	61.3	60.3
160	59.3	58.3	57.3	56.3	55.3	54.3	53.3	52.3	51.3	50.3
170	49.3	48.3	47.3	46.3	45.3	44.3	43.3	42.3	41.3	40.3
180	0	0	6140.2							
			*3							

* Radiosonde serial number

*1 Detent value

*2 Sum of first three digits of radiosonde serial number, the last four digits of radiosonde serial number, and the detent value.

*3 Checksum

TABLE 11 - AN/AMT-4 HUMIDITY TABLE
E[14,15]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	8	16	24	32	40	48	56	64	72	80	88	96	104	H/T
2	2745	2675	2601	2510	2360	2134	1861	1505	1010	422	91	-10	-28	-40
3	2737	2667	2604	2499	2374	2158	1904	1562	1090	495	115	10	-28	-32
4	2727	2665	2600	2512	2382	2188	1949	1631	1205	637	179	28	-28	-24
5	2726	2658	2599	2509	2390	2204	1983	1700	1320	824	307	88	-16	-16
6	2724	2652	2596	2508	2395	2222	2021	1757	1421	998	498	190	-6	-8
7	2719	2651	2595	2510	2401	2248	2052	1804	1499	1117	673	299	31	0
8	2712	2648	2591	2506	2408	2261	2079	1868	1558	1198	788	389	51	8
9	2708	2641	2588	2506	2411	2270	2107	1883	1608	1266	875	461	81	16
10	2704	2638	2585	2506	2418	2288	2130	1920	1650	1322	942	521	117	24
11	2692	2630	2580	2506	2422	2295	2154	1954	1689	1378	1004	582	156	32
12	2685	2625	2569	2506	2429	2306	2178	1985	1725	1428	1060	669	255	40
13	2685	2625	2569	2506	2429	2306	2178	1985	1725	1428	1060	669	255	48
14	2685	2625	2569	2506	2429	2306	2178	1985	1725	1428	1060	669	255	56
15	2685	2625	2569	2506	2429	2306	2178	1985	1725	1428	1060	669	255	64

Column 14 = dry bulb temperature ^{o}C
Row 1 = Relative Humidity in %
Rows 2-15 = Resistance of humidity in ohms

TABLE 12 - J005 HUMIDITY TABLE
E[14, 15]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	8	16	24	32	40	48	56	64	72	80	88	96	104	H/T
2	18659	17038	15740	14707	13086	10887	8007	4793	717	-4150	-8708	-13082	-17480	-40
3	18577	16968	15674	14722	13155	11092	8337	5298	1291	-3602	-8077	-12620	-17134	-32
4	18490	16885	15627	14731	13242	11308	8689	5805	1919	-2973	-7407	-12161	-16815	-24
5	18395	16796	15601	14737	13335	11532	9034	6285	2560	-2279	-6693	-11703	-16499	-16
6	18293	16704	15567	14732	13427	11778	9352	6750	3177	-1562	-5940	-11234	-16210	-8
7	18211	16628	15524	14701	13498	11967	9689	7211	3768	-907	-5157	-10773	-16077	0
8	18105	16558	15475	14694	13555	12153	9999	7619	4294	-290	-4387	-10281	-16037	8
9	18021	16487	15424	14682	13616	12327	10292	7985	4757	299	-3687	-9770	-16016	16
10	17963	16413	15373	14662	13680	12481	10555	8313	5159	837	-3078	-9258	-16000	24
11	17891	16346	15322	14648	13721	12618	10767	8604	5501	1267	-2547	-8778	-16030	32
12	17819	16339	15272	14634	13762	12768	10979	8900	5851	1697	-2000	-8298	-16000	40
13	17819	16339	15272	14634	13762	12768	10979	8900	5851	1697	-2000	-8298	-16000	48
14	17819	16339	15272	14634	13762	12768	10979	8900	5851	1697	-2000	-8298	-16000	56
15	17819	16339	15272	14634	13762	12768	10979	8900	5851	1697	-2000	-8298	-16000	64

Column 14 = Dry bulb temperature °C
Row 1 = Relative Humidity in %
Rows 2-15 = Resistance of humidity element in ohms

2. FILE DESCRIPTION

The Hewlett-Packard cartridge has two tracks of data. The files

a. Track Ø

		Section
File Ø:	Master Supervisory File	Command supervisor
File 1:	Functional key definitions	
File 2:	Date, Survey, Linearity, Pressure	Preflight
File 4:	A[*]	
File 5:	Baseline, Launch	
File 6:	Trunk	Flight
File 7:	W$, U$, V$, B[*]	
File 8:	K[*], E[*], X[*]	
File 9:	P$, Q$, O$, L$, N$, G$, D$	
File 10:	F[*], Y[*]	
File 11:	H[*]	
File 12:	Sound ranging, computer meteorology, fallout, ballistic 3	Output
File 13:	Ballistic 2	
File 14:	AWS First Transmission	
File 15:	Last flight update	
File 16:	AWS Second transmission	
File 3:	Not used	

b. Track 1

		Section
File 0:	W$, U$, V$, B[*]	P. TAPE
File 2:	P. TAPE	
File 1:	PIBAL (Not used)	PIBAL
File 3:	TACFIRE	TACFIRE
File 4:	Sound Ranging, Computer Met, Fallout, Ballistic 3	Last Flight
File 5:	P$, Q$, O$, L$, N$, G$, D$	
File 6:	F[*], Y[*]	
File 7:	H[*]	
File 8:	Ballistic 2	
File 10:	AWS First Transmission	
File 11:	AWS Second Transmission	
File 9:	INFORMATION	INFO

c. T list Files

The following is a t list of the files on track Ø and 1 showing the size of the files and the amount of bytes stored in each file.

T LIST TRACK Ø

trk Ø

#0	Bytes	Size
6	692	1000
#1		
5	800	1000
#2		
6	15628	20000
#3		
3	24	100
#4		
2	88	500
#5		
6	15358	20000
#6		
6	11032	12000
#7		
3	1272	1500
#8		
2	2064	2500
#9		
3	420	500
#10		
2	2992	5000
#11		
2	2304	3000
#12		
6	12876	15000
#13		
6	7612	13000
#14		
6	11478	15000
#15		
6	444	8000
#16		
6	7106	15000
#17		
0	0	0

T LIST TRACK 1

trk 1

#0	Bytes	Size
3	1272	5000
#1		
6	5106	10000
#2		
6	6272	10000
#3		
6	212	5000
#4		
6	13078	15000
#5		
3	420	500
#6		
2	2992	5000
#7		
2	2304	3000
#8		
6	7620	13000
#9		
6	2136	5000
#10		
6	11488	15000
#11		
6	7112	15000
#12		
0	0	0

3. VARIABLE DESCRIPTION

a. Array Variables

dim A[11] - Launch values

A[1] = Azimuth offset at launch A[11] = Flight Counter
A[2] = Horizontal distance offset
A[3] = Surface wind direction
A[4] = Surface wind speed
A[5] = Surface press at launch
A[6] = Corrected surface temperature
A[7] = Not used
A[8] = Corrected surface humidity
A[9] = Surface virtual temperature
A[10] = Preflight/continue flag
A 11 = Flight Counter

dim B[25] - Baseline values

B[1] = Thermistor resistance RT
B[2] = Temperature ordinate value "Temp Comp" input
B[3] = Low Temperature Reading
B[4] = Normalized temperature RD "Correcting RD" input
B[5] = RD correction at baseline "Temp Comp" output
B[6] = Recorder value above temperature ordinate "Correcting RD"
B[7] = Recorder value below temperature ordinate "Correcting RD"
B[8] = RD correction at B[6] "Correcting RD"
B[9] = RD correction at B[7] "Correcting RD"
B[10] = RD correction for normalized temperature RD "Correcting RD" output
B[11] = Baseline humidity constant
B[12] = Baseline temperature calibration constant
B[13] = Not used
B[14] = Last pressure contact
B[15] = Surface pressure
B[16] = Auto contact correction
B[17] = Pressure (higher) at whole contact below surface
B[18] = Contact at B[17]
B[19] = Pressure (lower) at whole contact above surface
B[20] = Contact at B[19]
B[21] = Not used
B[22] = Geopotential height constant for V$
B[23] = Pressure scaling factor for V$
B[24] = Index into V$ for surface
B[25] = Baseline humidity calibration constant

dim C[0:16, 1:14] - Ballistic weights and values

0:16 - zones where 0 - surface
1-15 - lines

C[N,1] = Zone level number
C[N,2] = Wind speed
C[N,3] = Temperature
C[N,4] = Pressure
C[N,5] = Wind azimuth
C[N,6] = % of standard temperature
C[N,7] = Weighted ballistic 3 temperature
C[N,8] = Weighted ballistic 2 temperature
C[N,9] = Densities
C[N,10] = % of standard density
C[N,11] = Weighted ballistic 3 densities
C[N,12] = Weighted ballistic 2 densities
C[N,13] = Weighted ballistic 3 or 2 X wind component
C[N,14] = Weighted ballistic 3 or 2 Y wind component

dim E[14,15] - Humidity tables

Table 11 - AN/AMT-4 Radiosonde
Table 12 - JOO5 Radiosonde

dim F[-1:60, 1:6] - Significant level data

at -1 level

#1	#2	#3	#4	#5	#6
Azimuth offset	Horizontal distance	First Missing Mandatory	Second Missing Mandatory	Third Missing Mandatory	Fourth Missing Mandatory

at 0 level
1-60 levels

#1	#2	#3	#4	#5	#6
Temperature °C	Humidity %	Temp virt °K	Pressure mb	Time min	Geopotential height meters

dim H[-2:45, 1:6] - Zone height data

at -2 level

#1	#2	#3	#4	#5	#6
				SFC Wind Direction	SFC Wind Speed

at -1 level

#1	#2	#3	#4	#5	#6
	SFC pressure		SFC temp Tv °K	Azimuth Offset	Horizontal distance

at 0-45 levels

#1	#2	#3	#4	#5	#6
Geometric height	log press	miscellan-eous	top of zone time	top of zone azimuth angle	top of zone elevation angle

Miscellaneous

H[0,3] = Acceleration of Gravity Ratio Constant
H[1,3] = P = zone level
H[2,3] = T = significant level
H[3,3] = Missing angles indicator
H[4,3] = Angle time interval (0.1, 0.5, 1.0)
H[5,3] = Last correct time of angles
H[6,3] = Last correct elevation angle
H[7,3] = Last correct azimuth angle
H[8,3] = Time below zone
H[9,3] = Elevation angle below zone
H[10,3] = Azimuth angle below zone
H[11,3] = Time above zone
H[12,3] = Elevation angle above zone
H[13,3] = Azimuth angle above zone
H[14,3] = Reference time below zone
H[15,3] = Reference time above zone

dim K[23,2] - Humidity Recorder Divisions (RD) Tables

TABLE 13: Humidity RD Table for AN/AMT-4 Radiosonde

Hum RD	Value	Hum RD	Value	Hum RD	Value
0	-735	32	1289	64	2016
4	0	36	1384	68	2119
8	384	40	1479	72	2232
12	625	44	1568	76	2359
16	800	48	1645	80	2509
20	945	52	1741	84	2692
24	1070	56	1830	88	2952
28	1184	60	1919		

(Values of resistance represent the difference between measured resistance at any altitude and the baseline resistance.)

TABLE 14: Humidity RD Table for J005 Radiosonde

Hum RD	Value	Hum RD	Value	Hum RD	Value
0	-28700	32	-4967	64	3394
4	-22700	36	-3846	68	4567
8	-17700	40	-2770	72	5831
12	-13700	44	-1736	76	7234
16	-10962	48	-748	80	8852
20	-9017	52	245	84	10977
24	-7522	56	1264	88	13000
28	-6180	60	2298		

(Values of resistance represent the difference between the measured resistance at any altitude and the baseline resistance.)

dim M[3,45] - AWS wind data

N - zone level

M[1,N] - height for zone N
M[2,N] - wind direction for zone N
M[3,N] - wind speed for zone N

dim X[2] - Calibration factors
X[1] - Baseline temperature
X[2] - Baseline humidity

b. String Variables

A$, B$, C$ - string areas
D$[16] - date of flight

Date-Time Group Format:

DDØTTTTZØMMMØYYØ

DD - numerical day of month (01 to 31)
TTTT - time value (0001 to 2400)
MMM - alphabetic designation of month (JAN-DEC)
YY - year (00 - 99)
Ø - blank

E$ [16] - string array for linearity
F$ [15] - wind data strings
G$ [192] - recorder check linearity data
H$ [2,32] - Baudot conversion array
L$ [16] - recorder check date-time group
N$ [6] - Recorder serial number
O$ [16] - previous survey date-time group

dim P$[102] - Header array

	1	2	3	4	5	6	7	8	9	10	11	12	
P$ [1,12] =	M	E	T	C	M	Q	A	A	A	O	O	O	)Computer meteor-
P$ [13,24]	Y	Y	G	G	G	V	H	H	H	P	P	P	)ological message.
P$ [25,36]	M	E	T	B	3	Q	A	A	A	O	O	O	NATO 3
P$ [37,48]	M	E	T	B	2	Q	A	A	A	O	O	O	NATO 2
P$ [49,60]	M	E	T	F	M	Q	A	A	A	O	O	O	Fallout
P$ [61,72]	Y	Y	G	G	G	H	H	H	H	T	R	O	
P$ [73,84]	M	E	T	S	R	Q	A	A	A	O	O	O	Sound Ranging
P$ [85,99]	Y	Y	G	G	G	V	±	T	T	T	D	D	D F F
P$ [100,100]	Y - AWS / N - No AWS												
P$ [101,101]	Y - 5-channel tape /N - 8-channel tape												
P$ [102,102]	4 - AN/AMT-4 Radiosonde/B - J005 WB Radiosonde												

dim Q$ [16] - AWS Data

Q$ [1,5] - AWS Location code
Q$ [6,13] - trending data for 1000 and 850 mb levels
Q$ [14,16]- not used

dim U$ [895] - Standard Pressure Table
See Table 10 for pressure values

dim V$ [96] - 15 contact pressures

dim W$ [56] - Mandatory contact levels (AWS)

Low altitude mbs									
1000	850	700	500	400	300	250	200	150	100

High altitude mbs									
70	50	30	20	10	7	5	3	2	1

c. r - variables

Extensive use of r-variables, r1 through r15, is made in the ballistic messages. These variables hold the weighting factors that are applied to density, temperature, and wind components. They represent the zone levels 1 to 15.

4. FLOW CHARTS.

a. System.

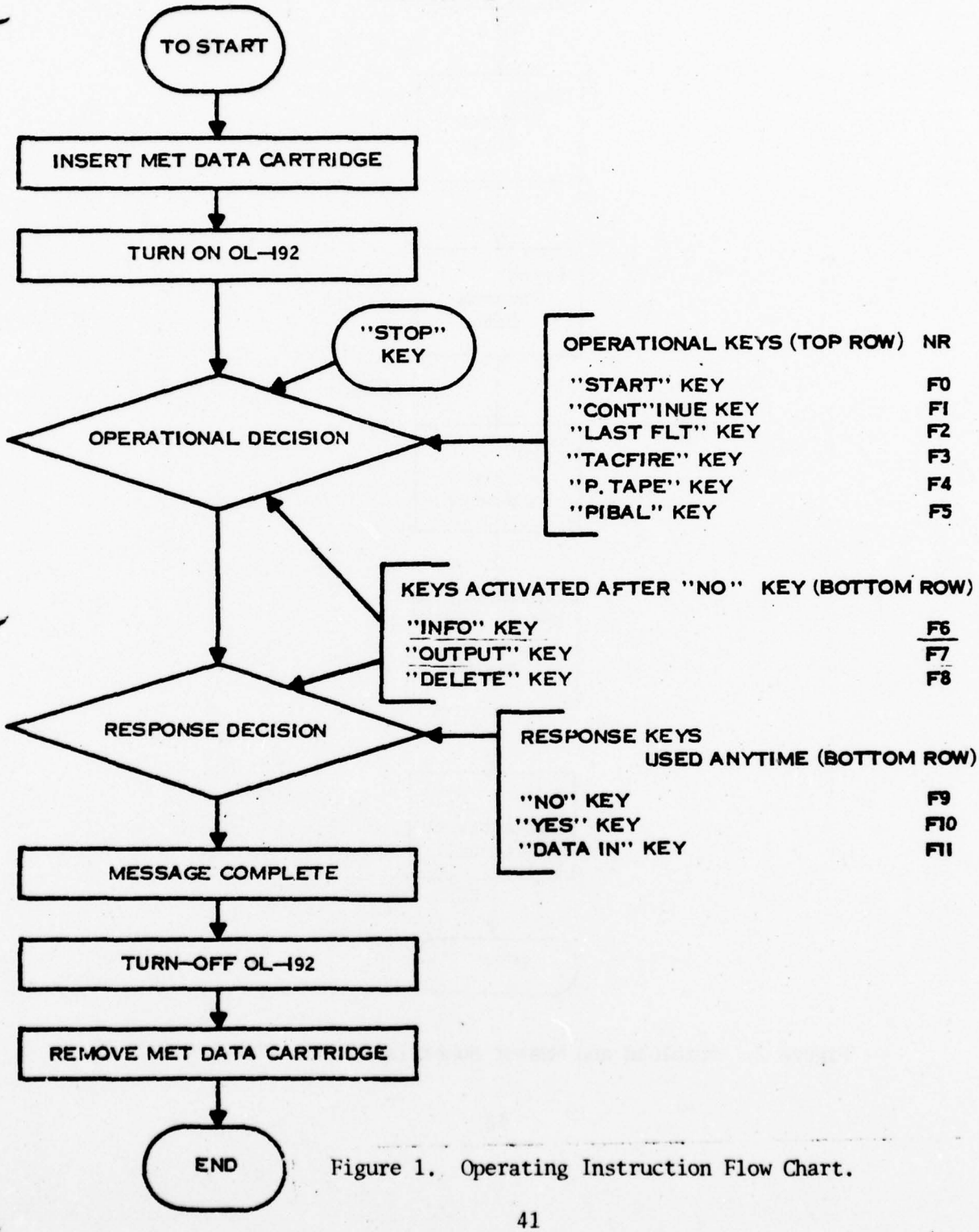


Figure 1. Operating Instruction Flow Chart.

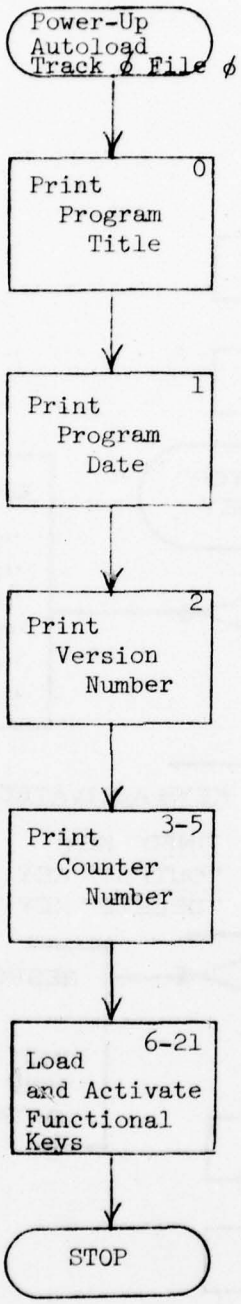


Figure 2. Autoload and Master Supervisory File (Track 0 File 0)

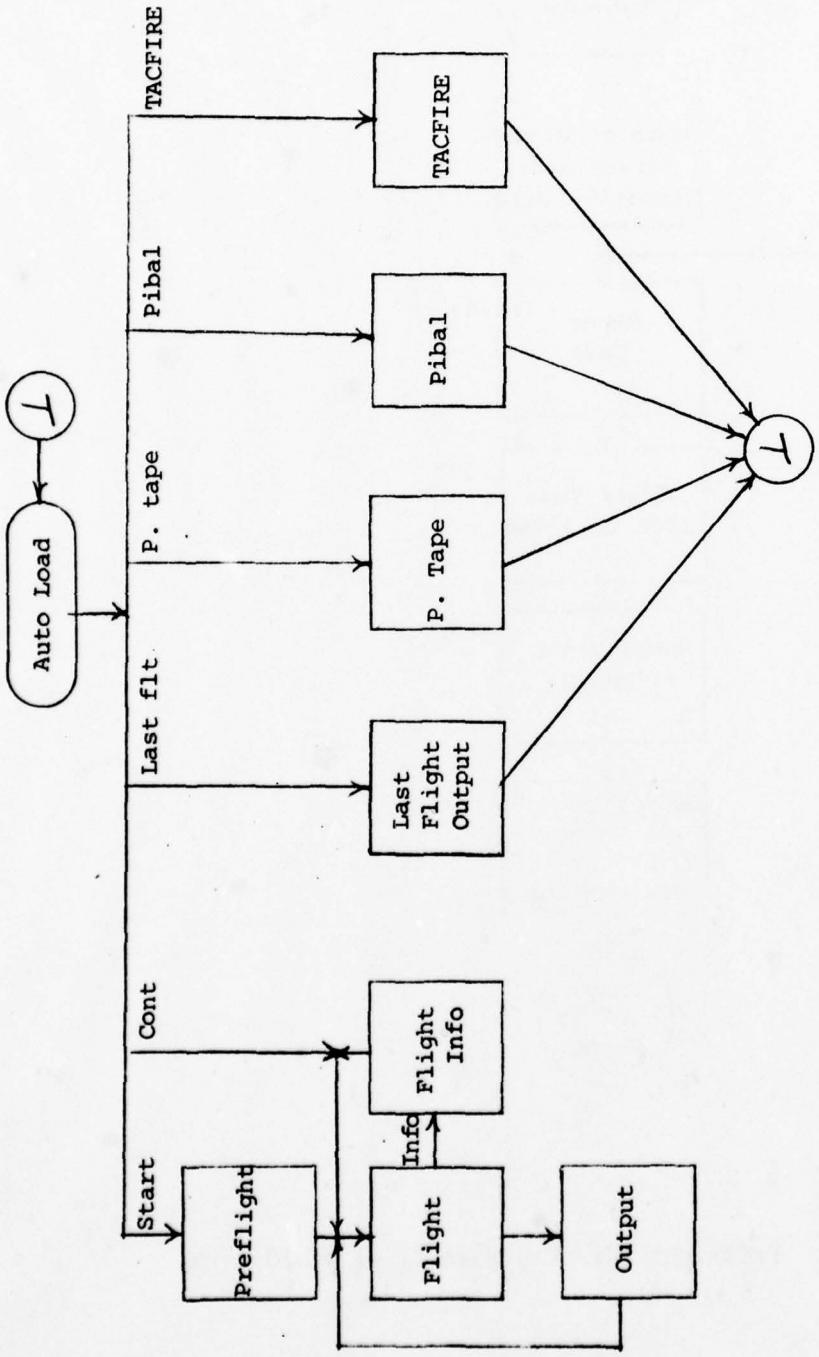


Figure 3. Functional Keys and Program Flow.

b. Preflight.

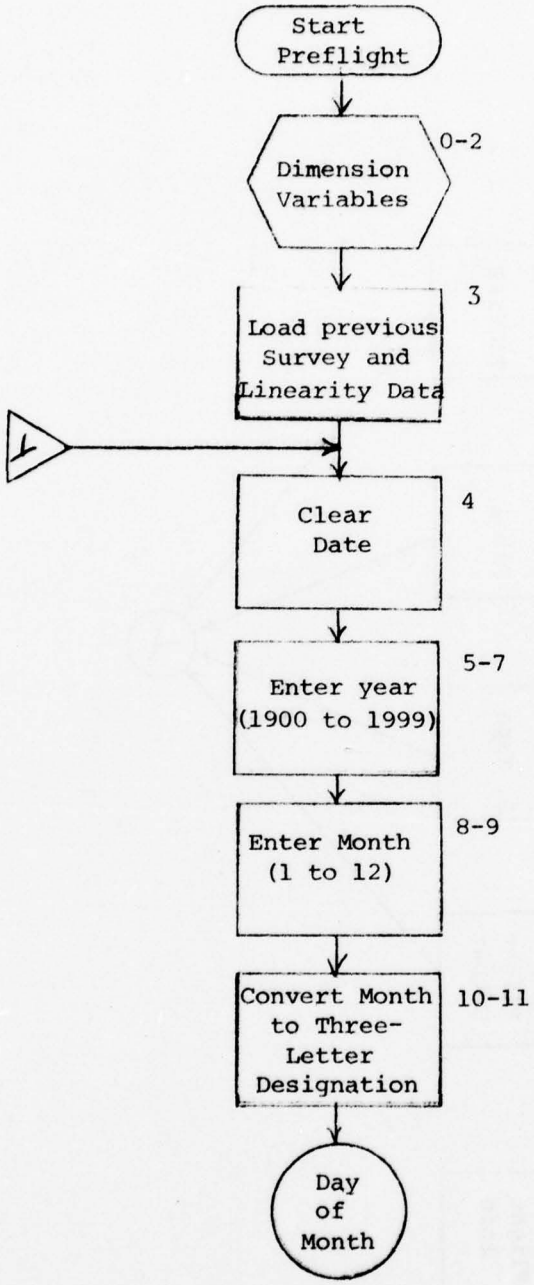


Figure 4a. Preflight (Track 0 File 2) (1 of 10)

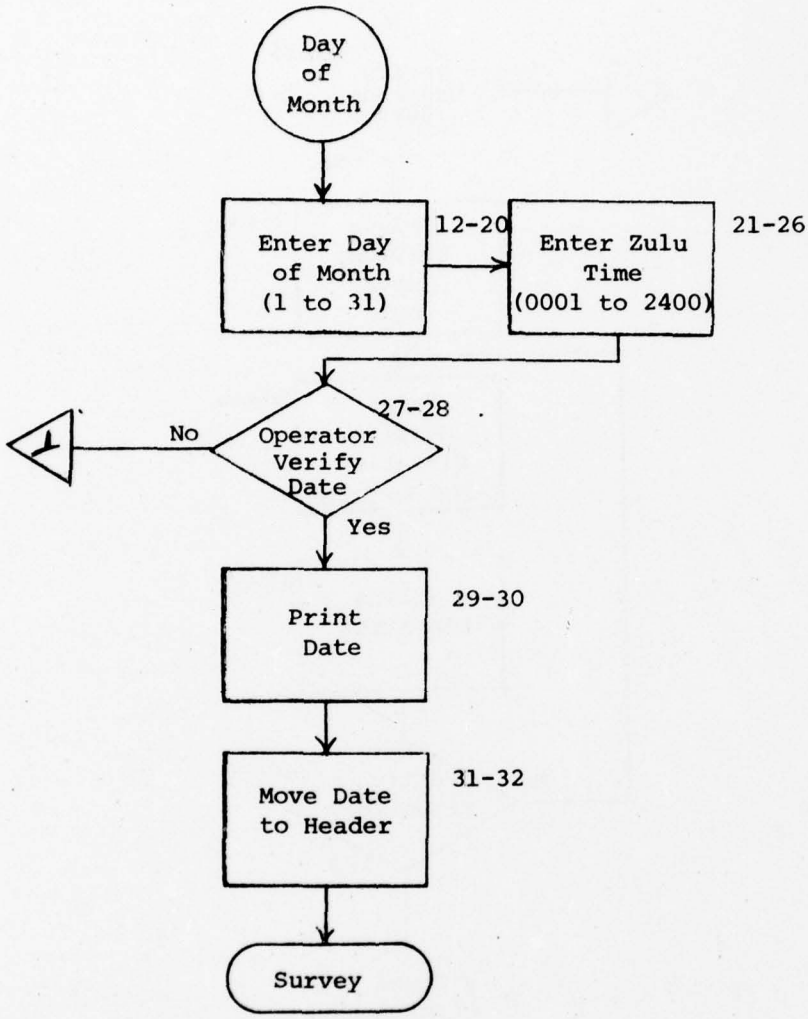


Figure 4b. Preflight Continued (2 of 10)

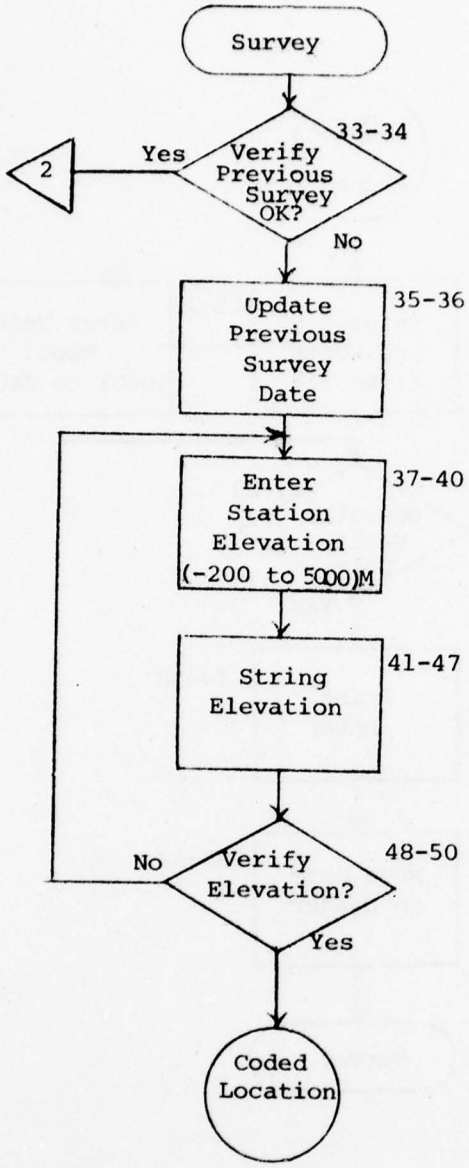


Figure 4c. Preflight Continued (3 of 10)

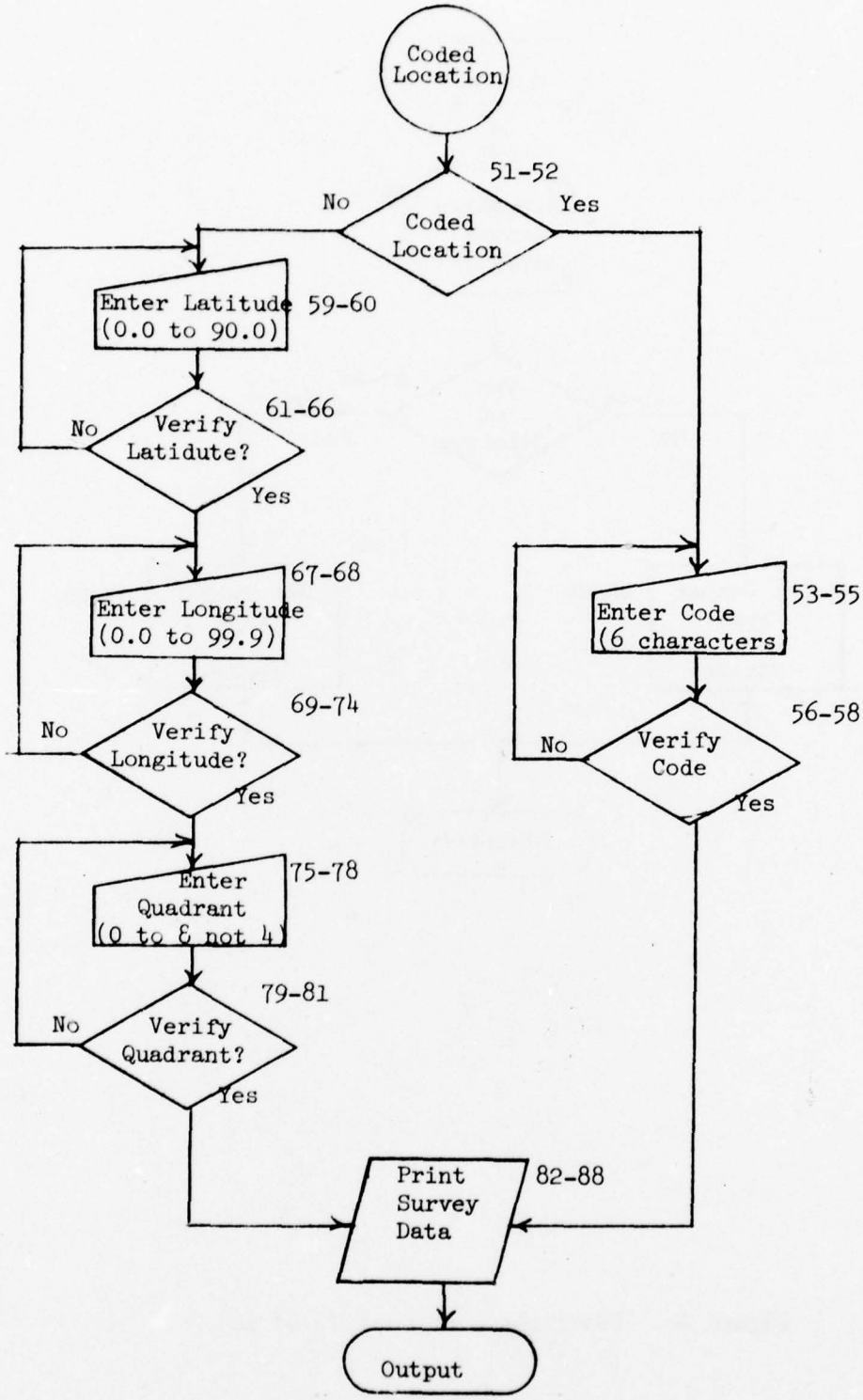


Figure 4d. Preflight Continued (4 of 10)

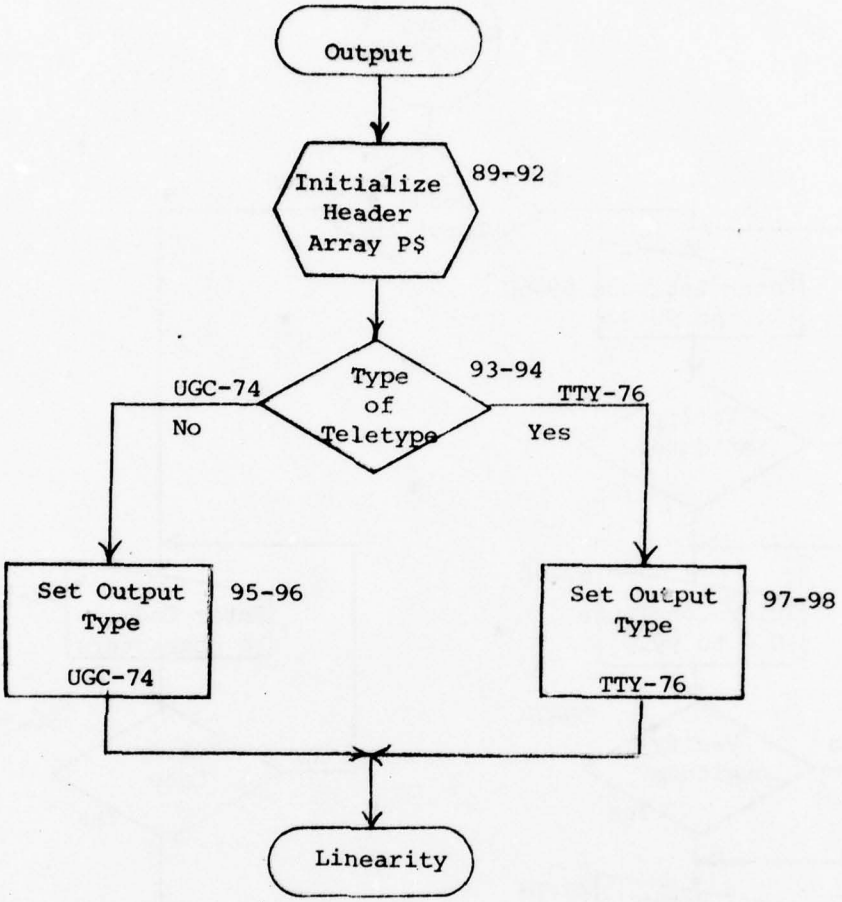


Figure 4e. Preflight Continued (5 of 10)

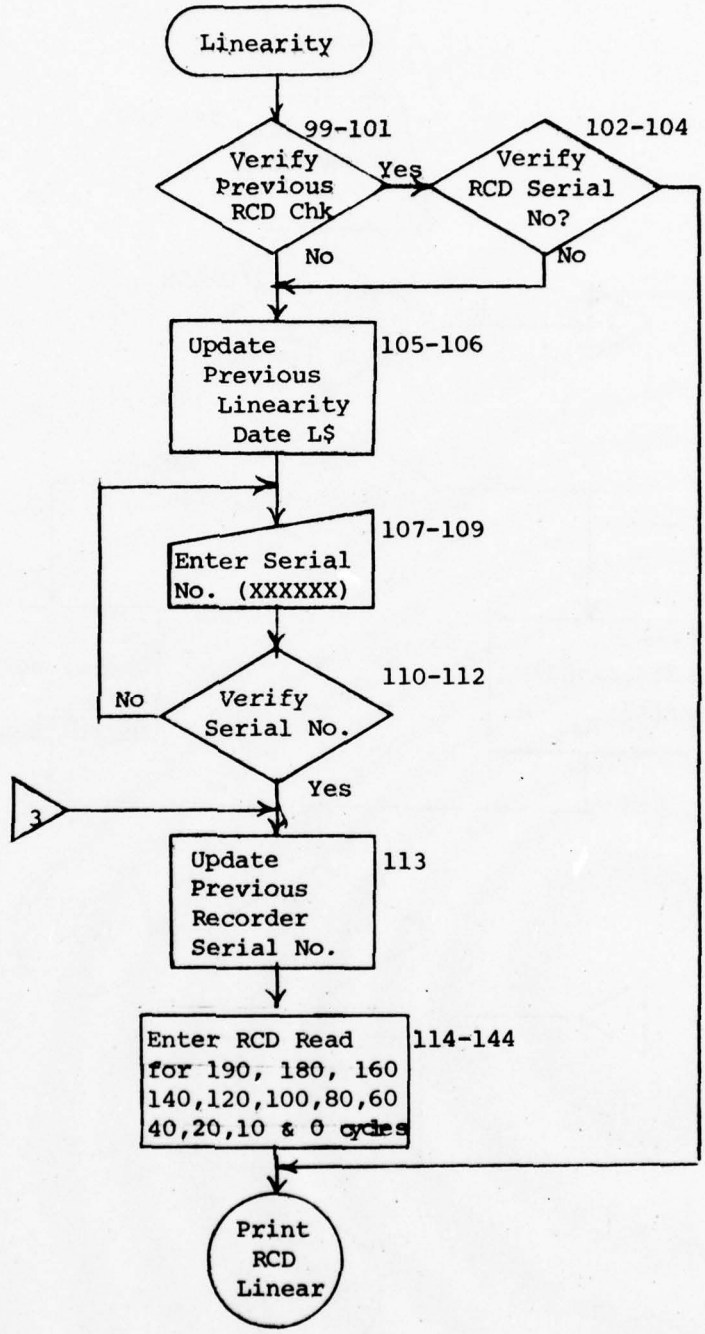


Figure 4f. Preflight Continued (6 of 10)

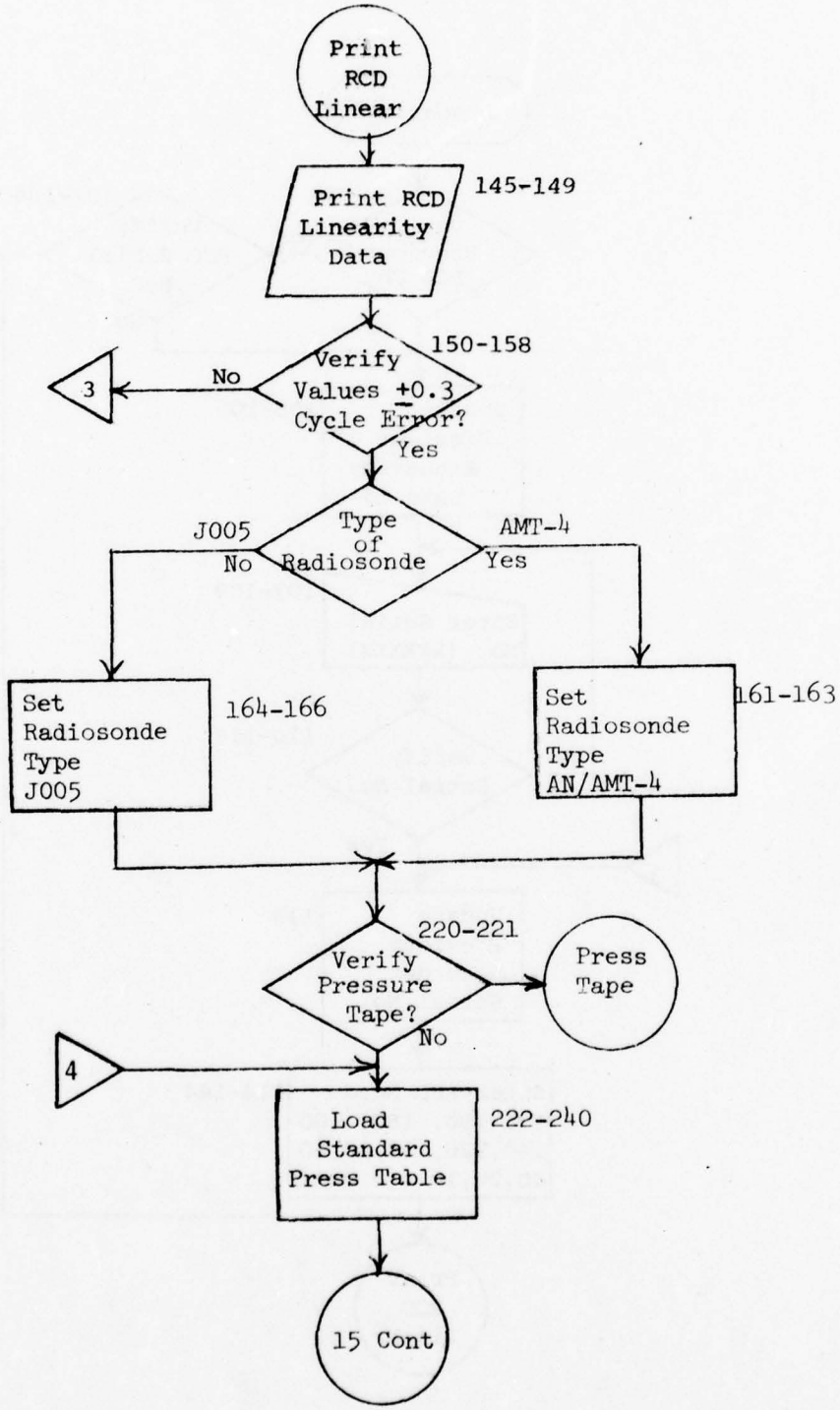


Figure 4g. Preflight Continued (7 of 10)

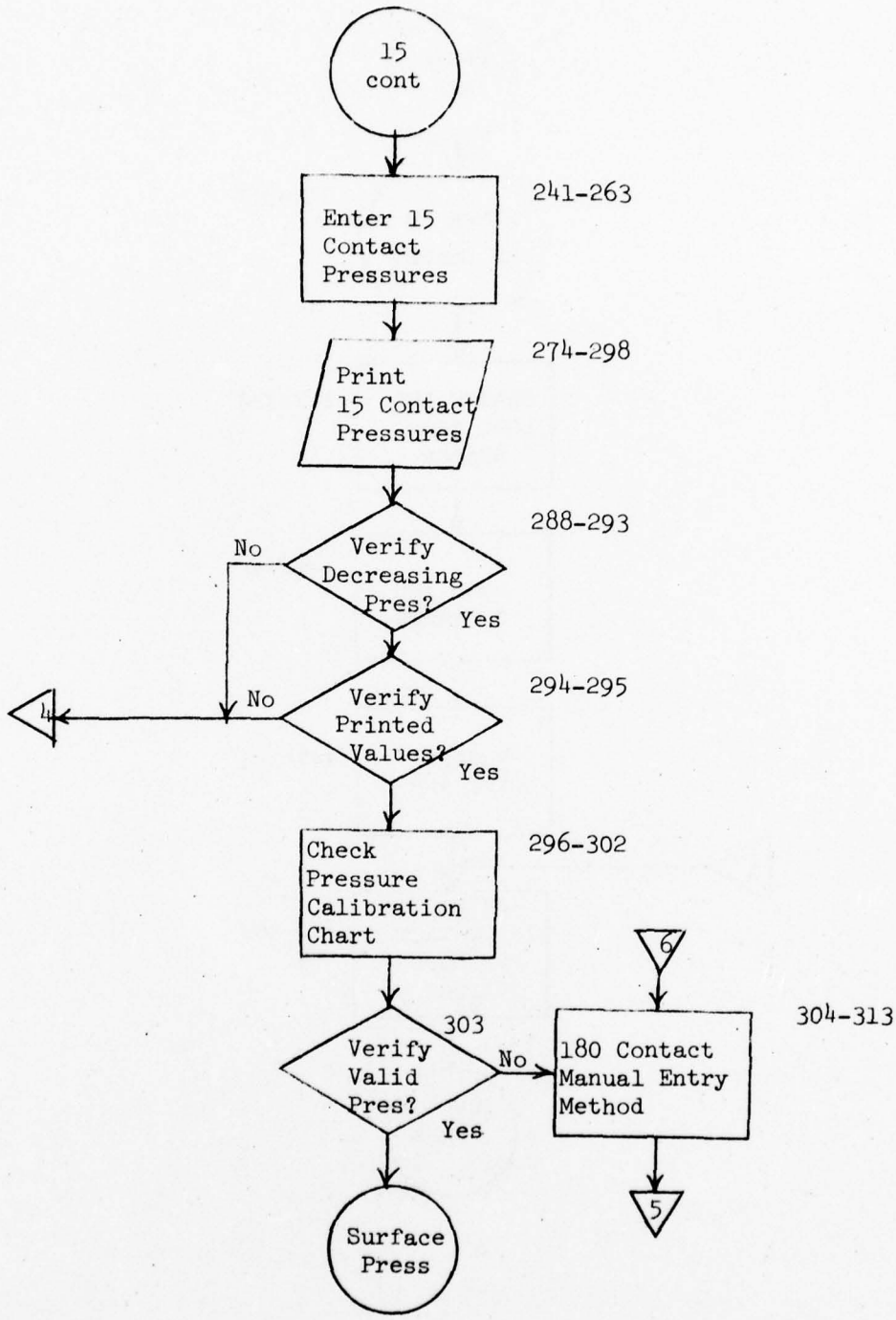


Figure 4h. Preflight Continued (8 of 10)

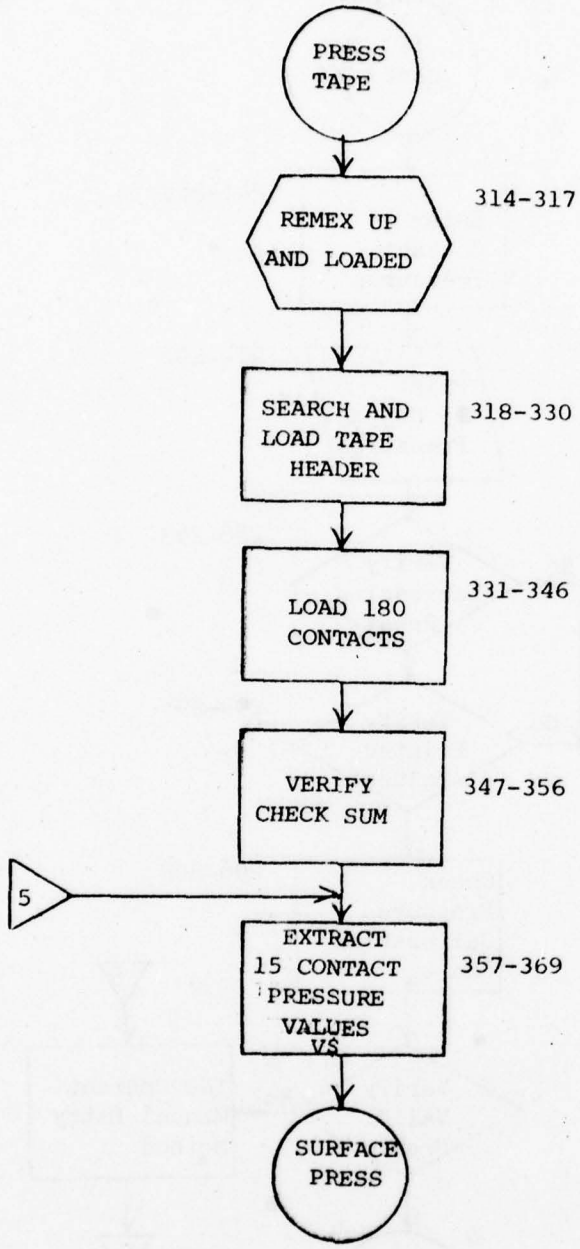


Figure 4i. Preflight Continued (9 of 10)

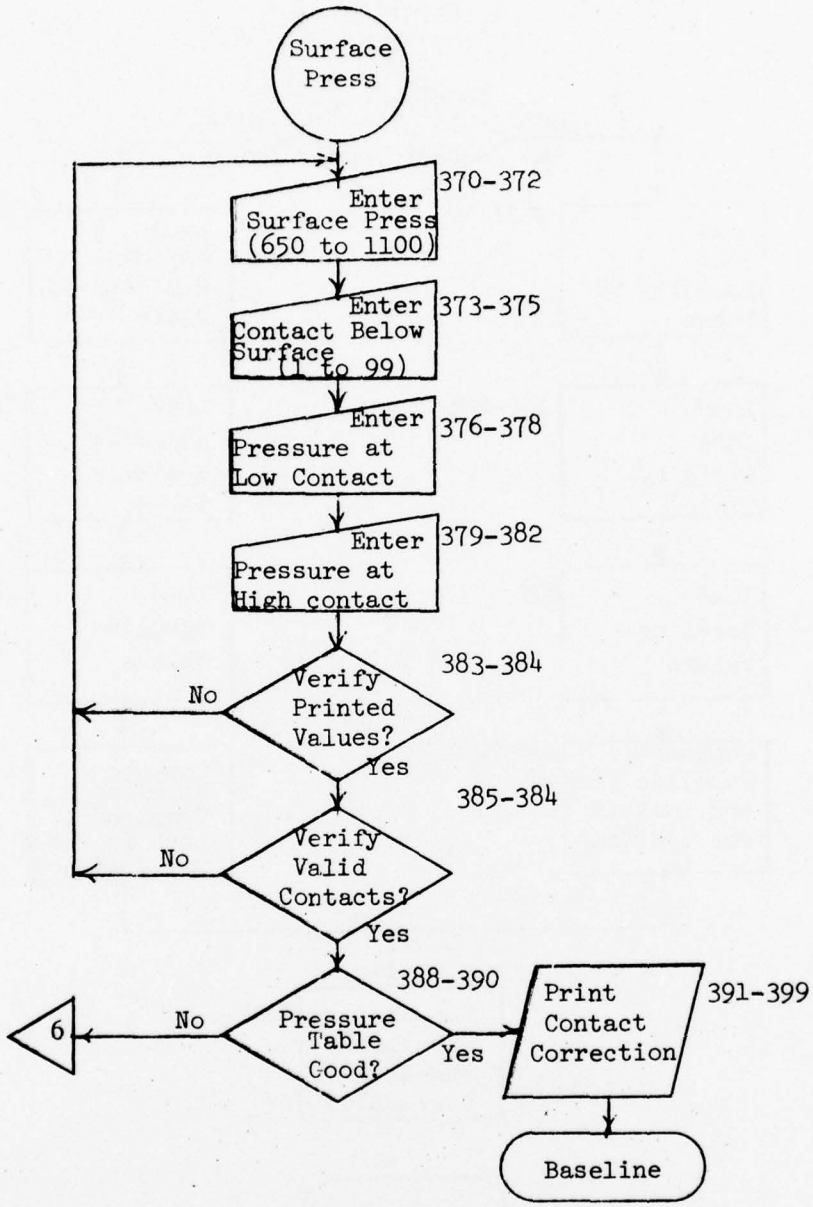


Figure 4j. Preflight Continued (10 of 10)

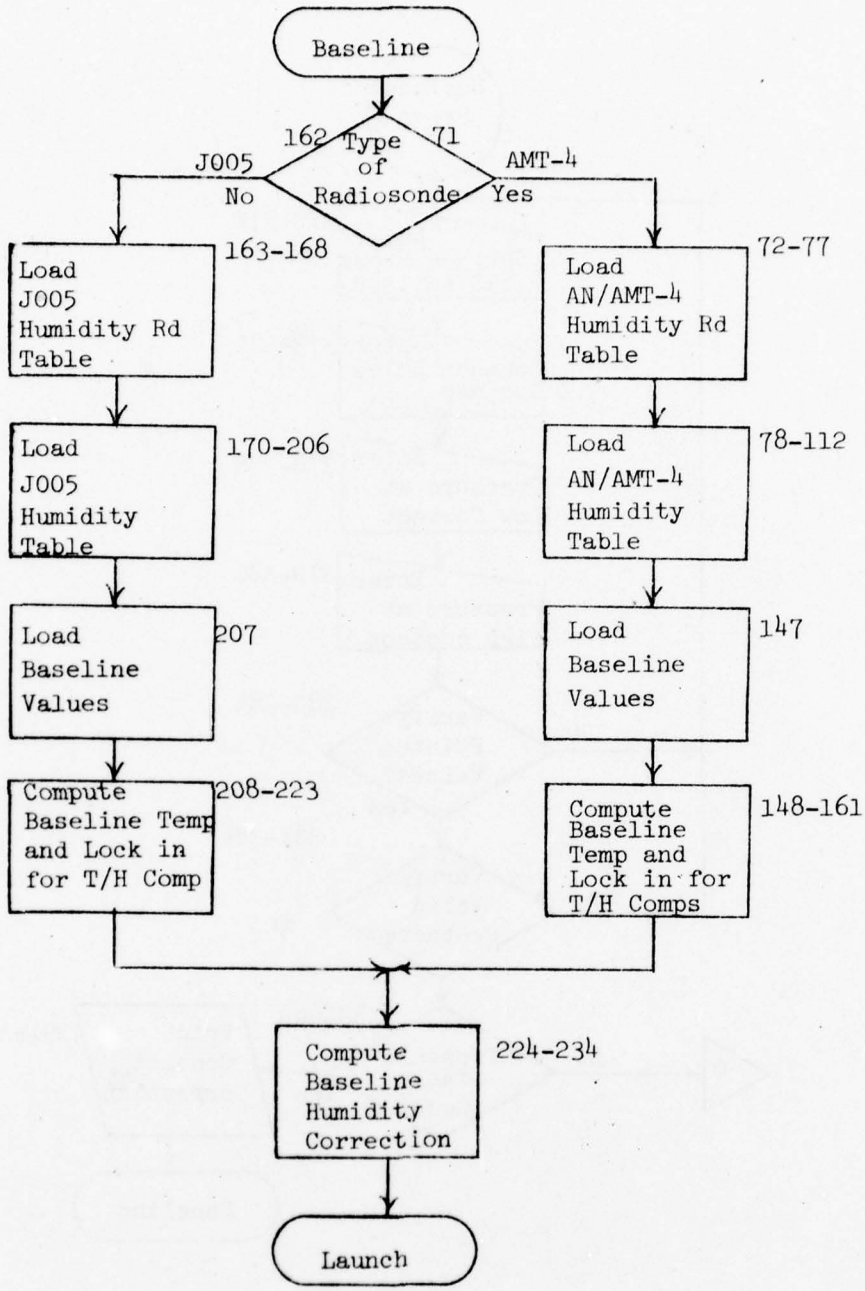


Figure 4k. Preflight Continued
(Track 0 File 5) (1 of 5)

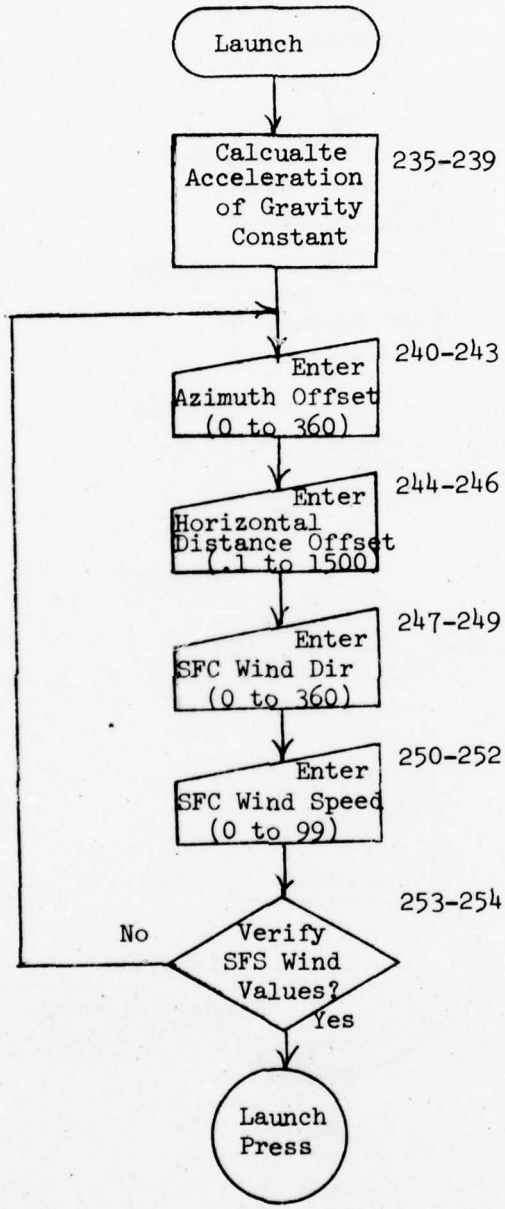


Figure 41. Preflight Continued (2 of 5)

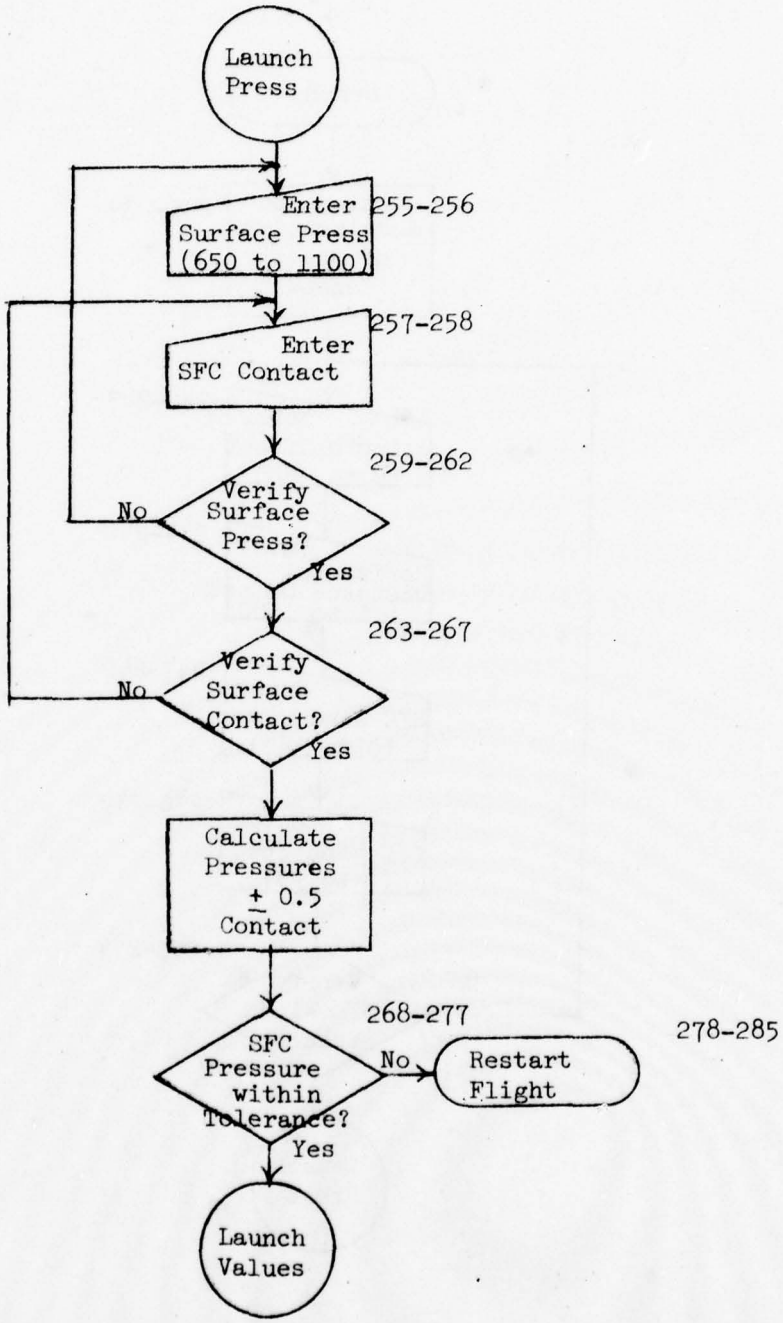


Figure 4m. Preflight Continued (3 of 5)

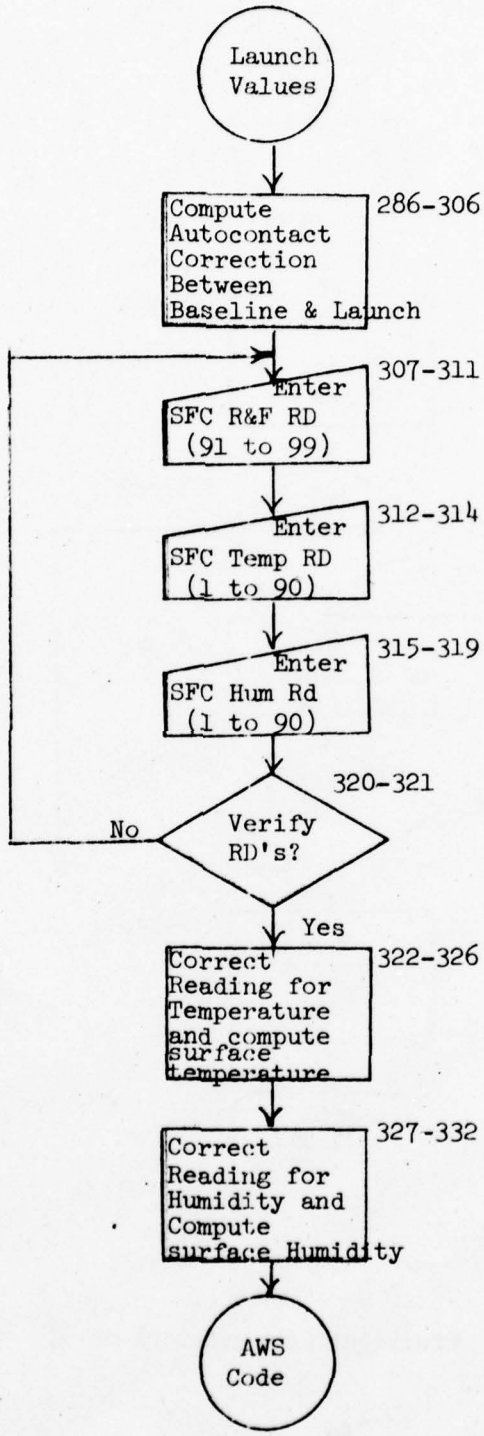


Figure 4n. Preflight Continued (4 of 5)

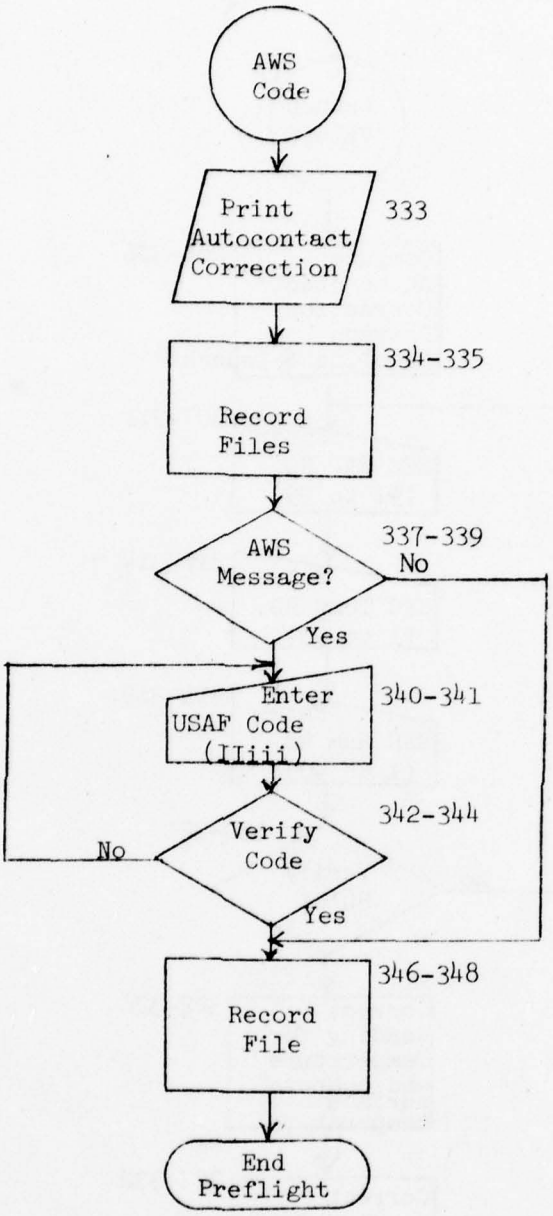


Figure 4o. Preflight Continued (5 of 5)

c. Flight.

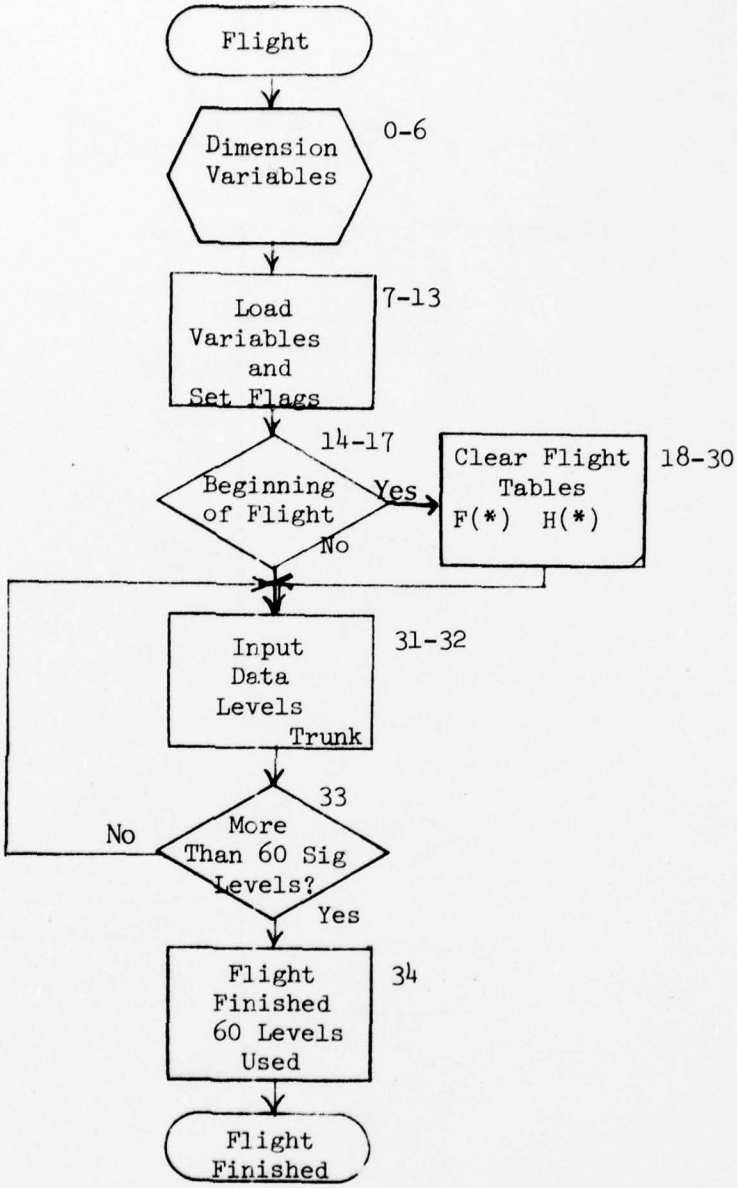


Figure 5a. Flight (Track 0 File 6) (1 of 5)

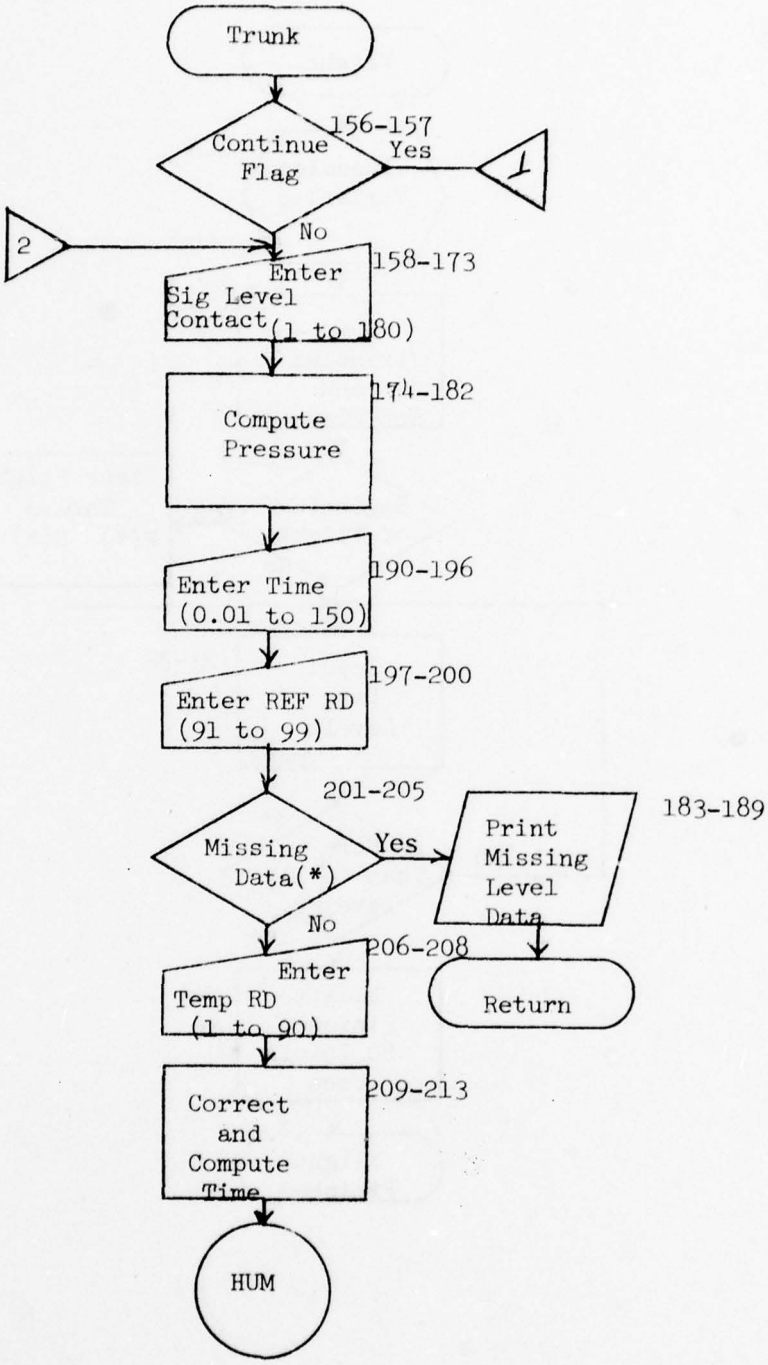


Figure 5b. Flight Continued (2 of 5)

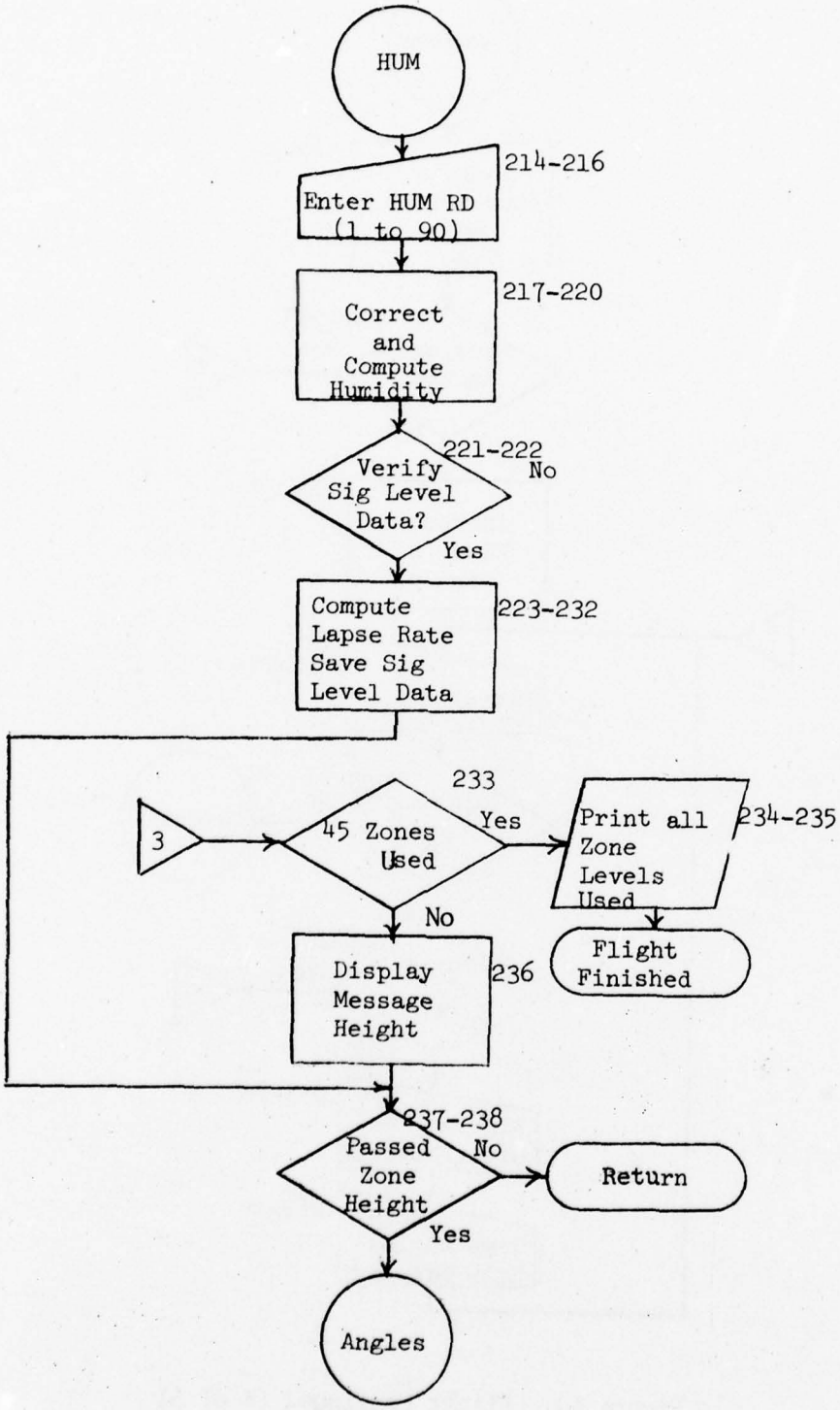


Figure 5c. Flight Continued (3 of 5)

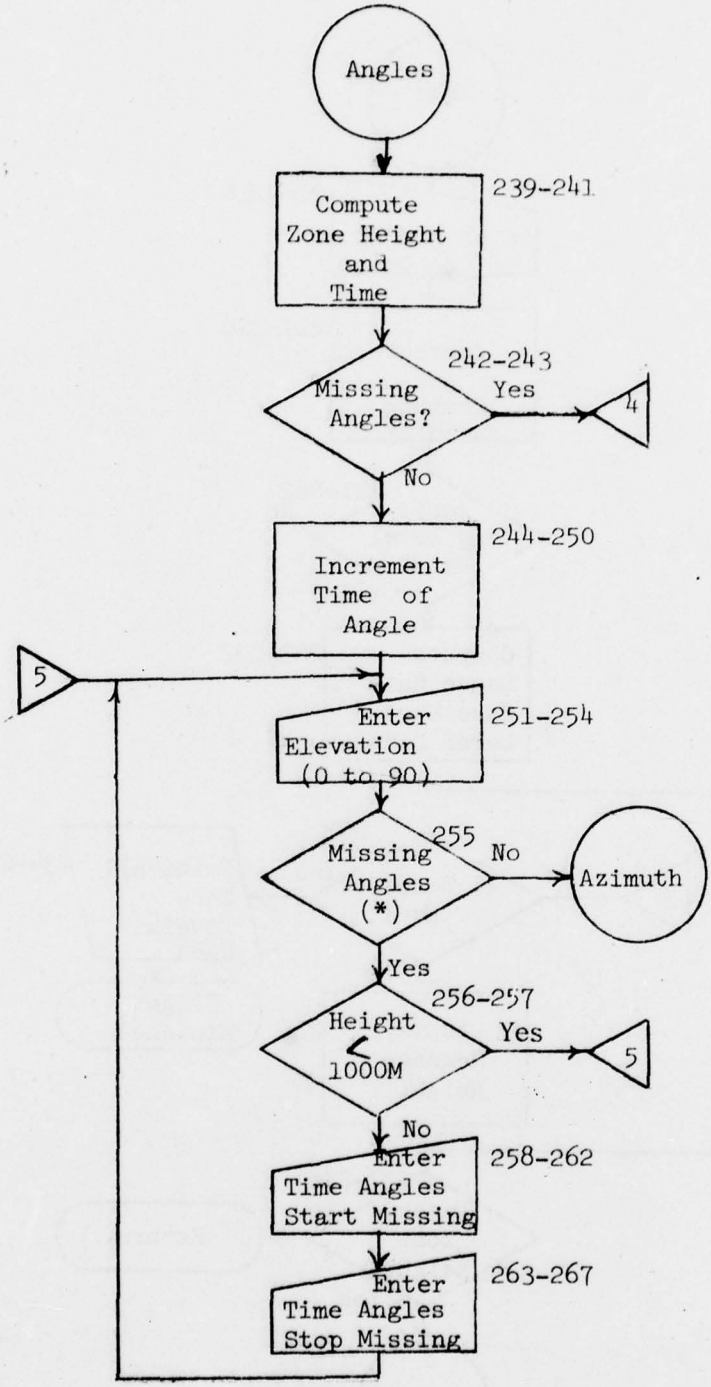


Figure 5d. Flight Continued (4 of 5)

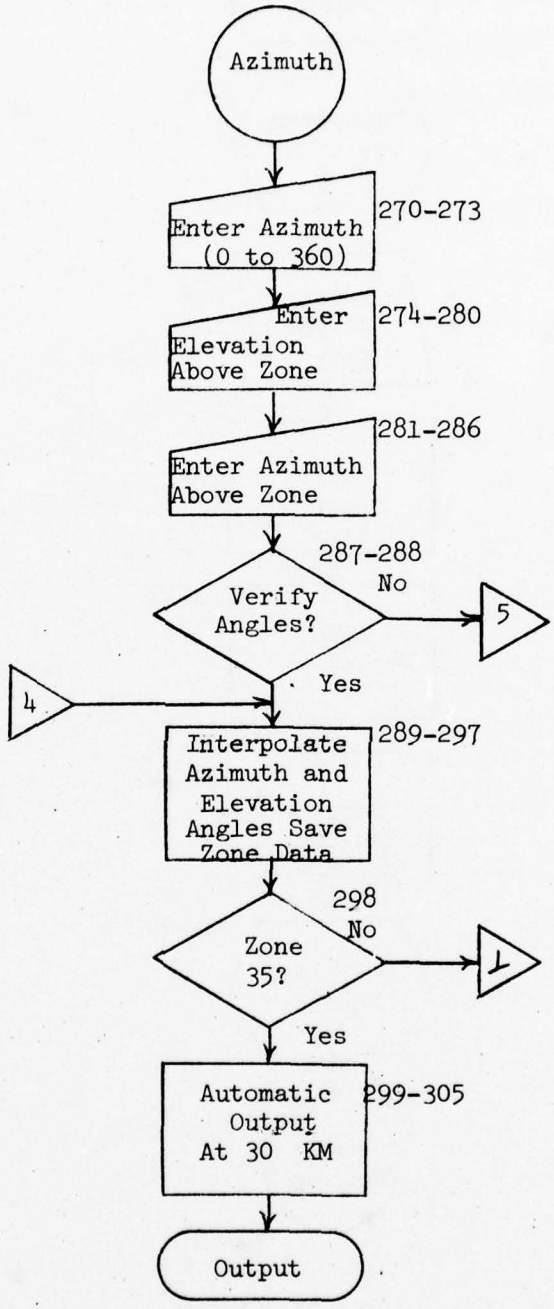


Figure 5e. Flight Continued (5 of 5)

d. Output.

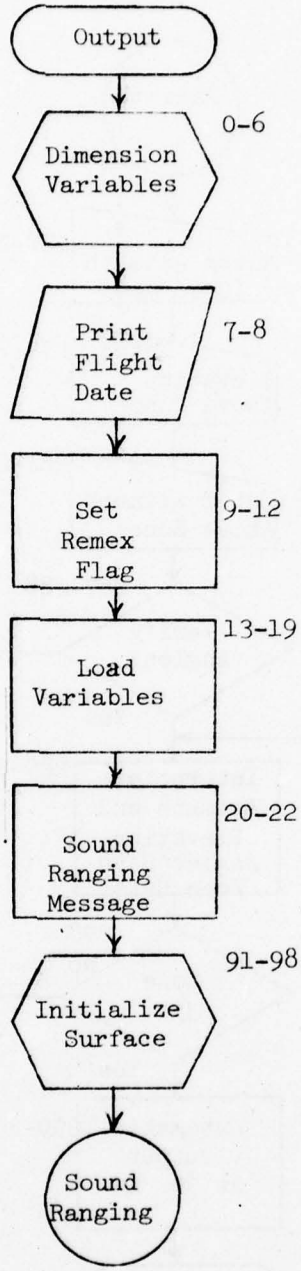


Figure 6a. Output Sound Ranging, Fallout Ballistic Met 3 (Track 0 File 12) (1 of 5)

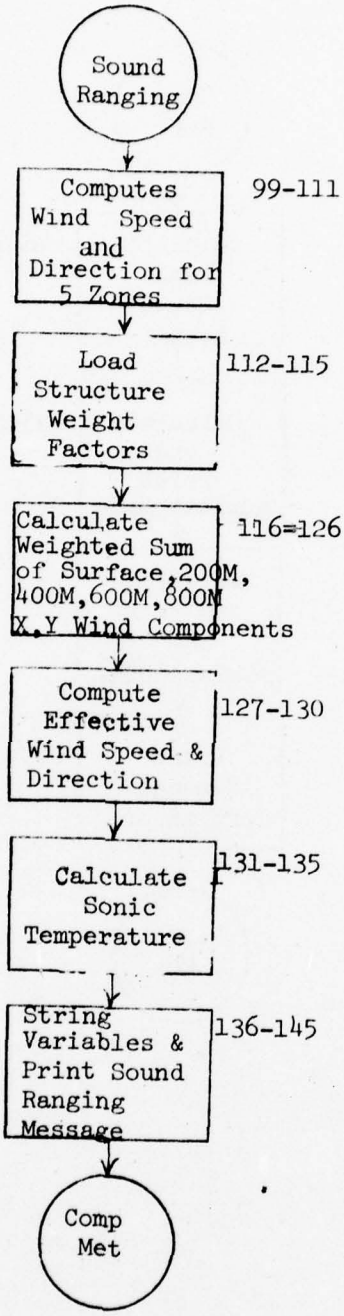


Figure 6b. Output Continued (2 of 5)

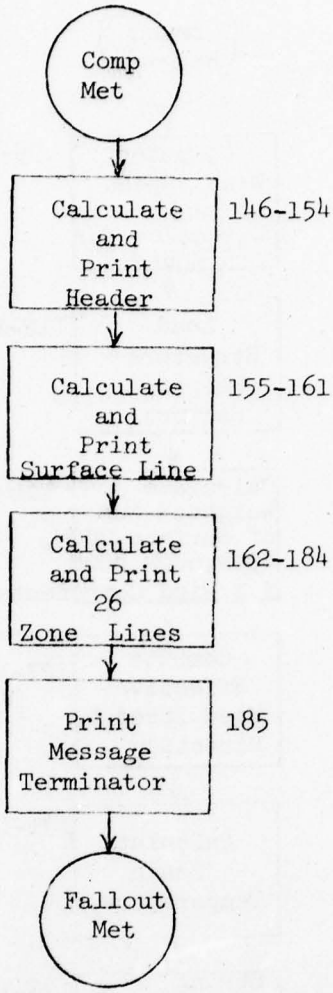


Figure 6c. Output Continued (3 of 5)

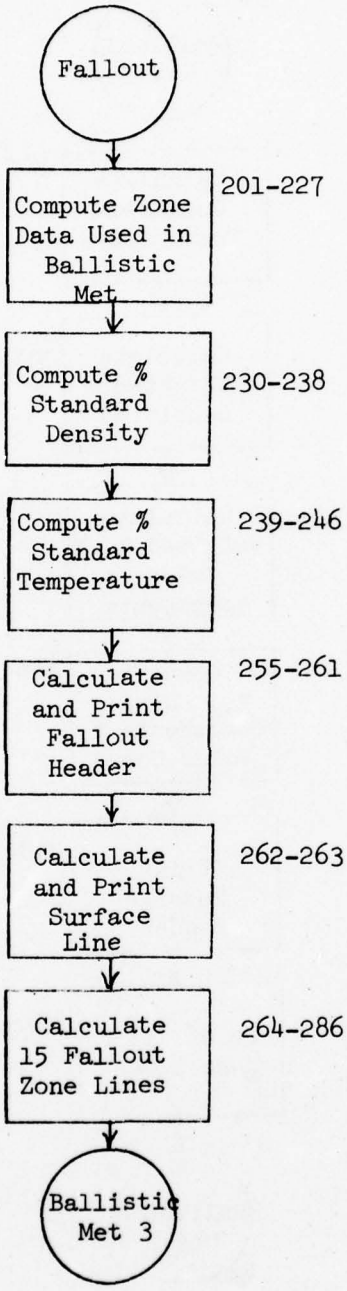


Figure 6d. Output Continued (4 of 5)

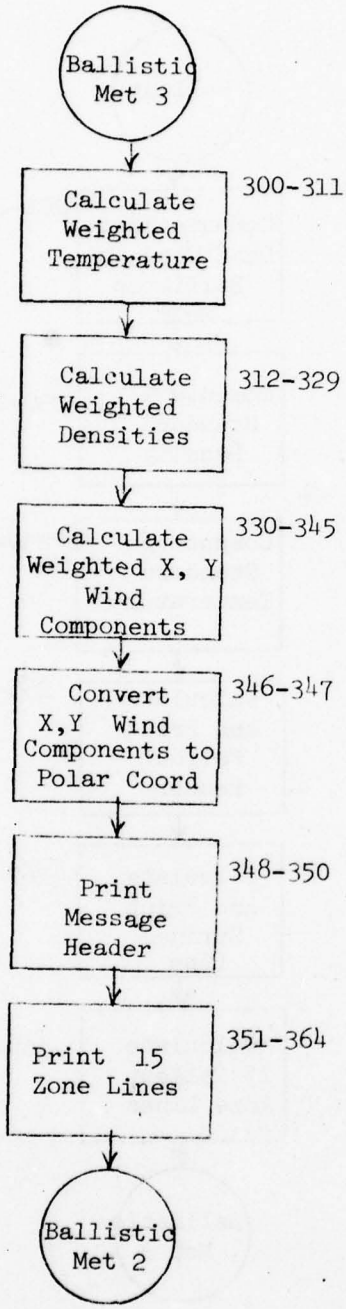


Figure 6e. Output Continued (5 of 5)

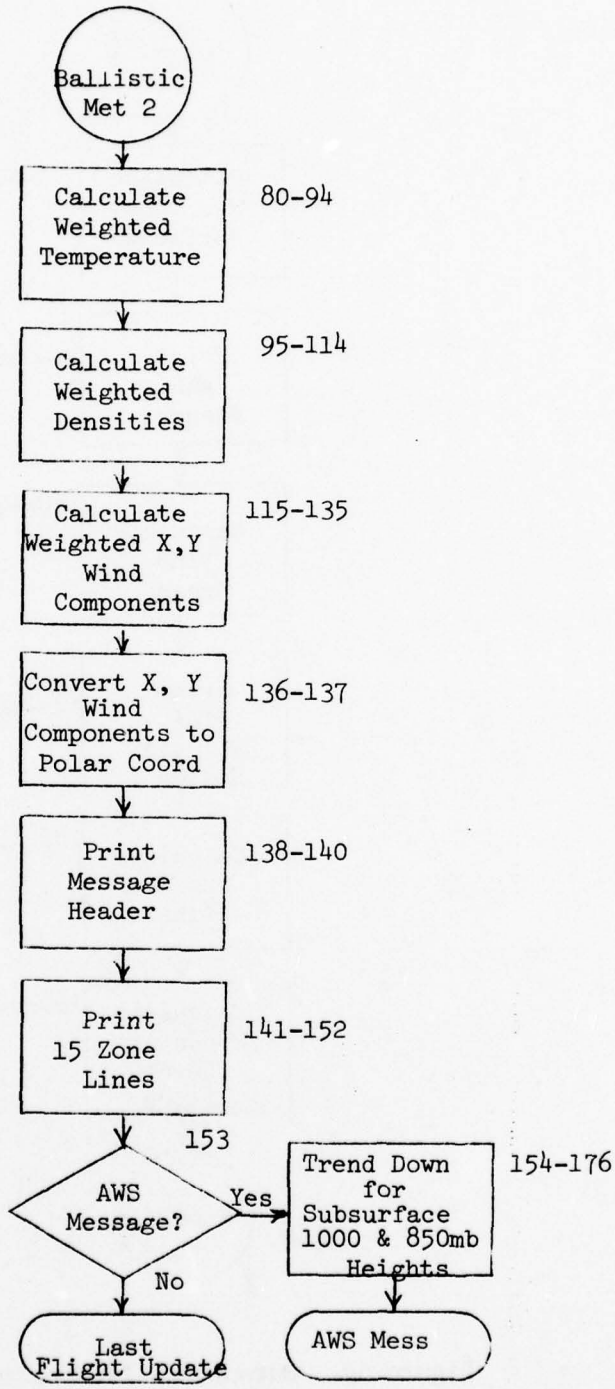


Figure 6f. Output Ballistic Met 2 (Track 0 File 13)

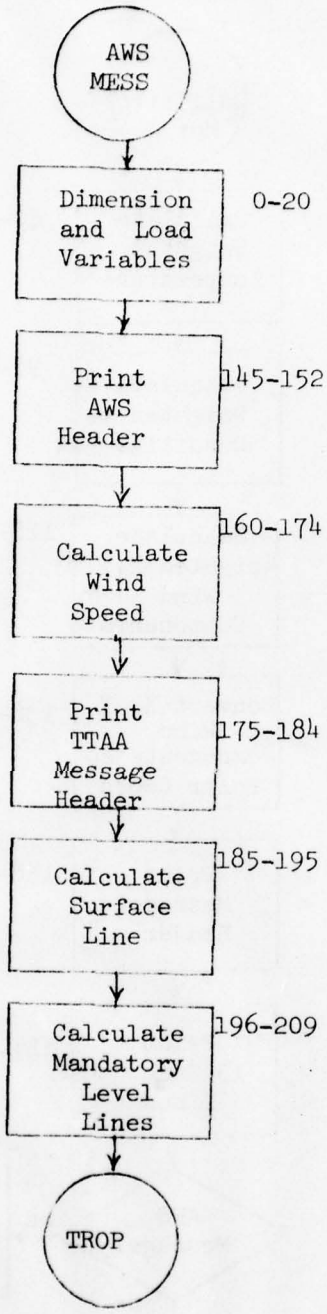


Figure 6g. Output AWS 1st Transmission
(Track 0 File 14) (1 of 3)

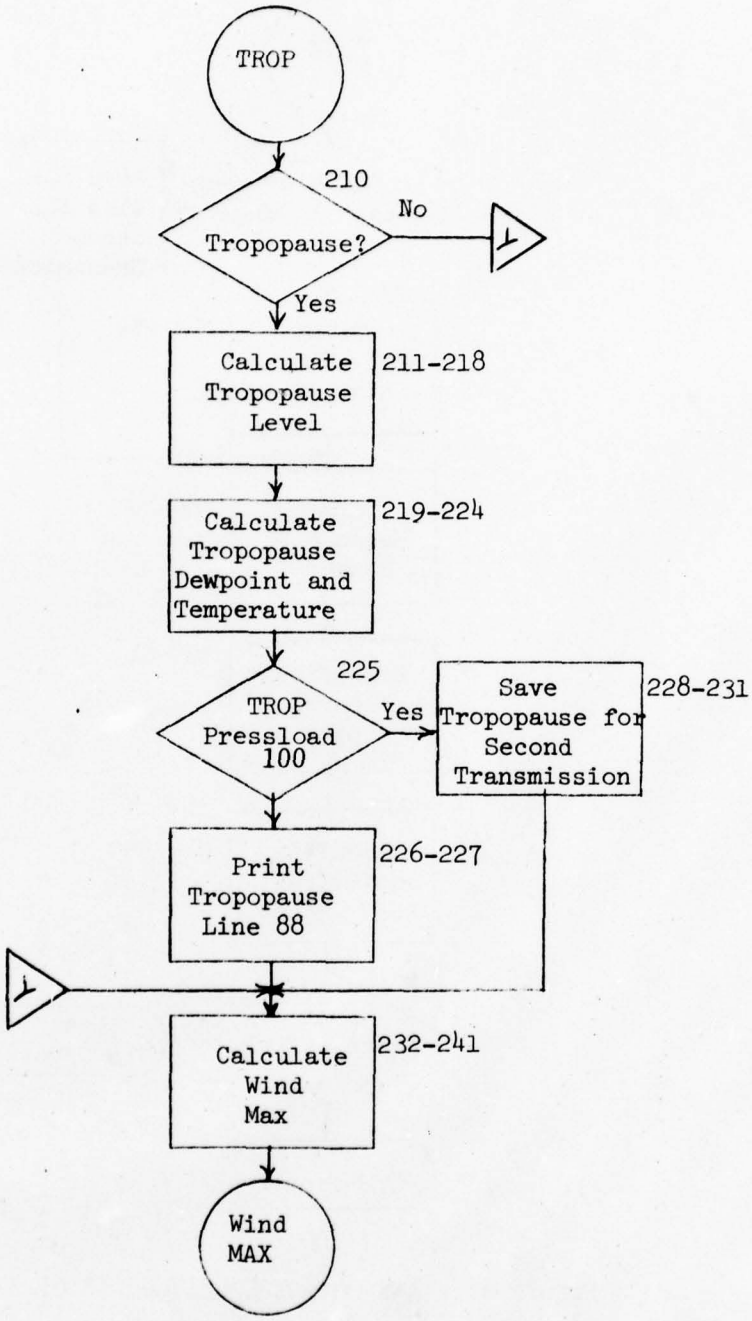


Figure 6h. AWS Output Continued (2 of 3)

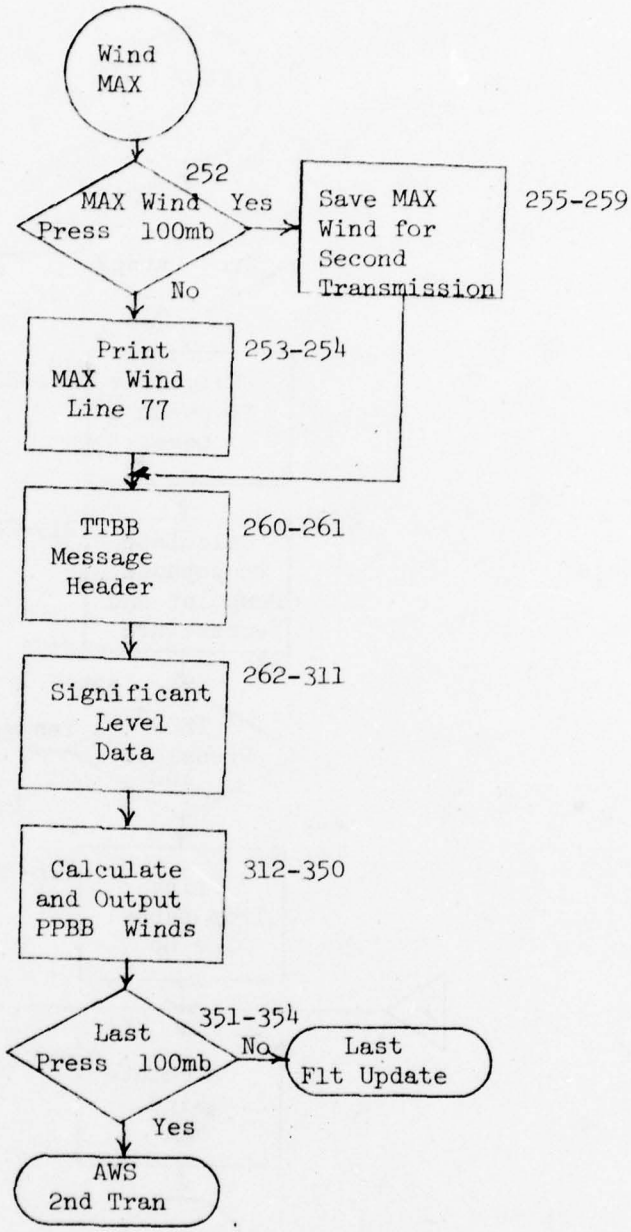


Figure 6i. AWS Output Continued (3 of 3)

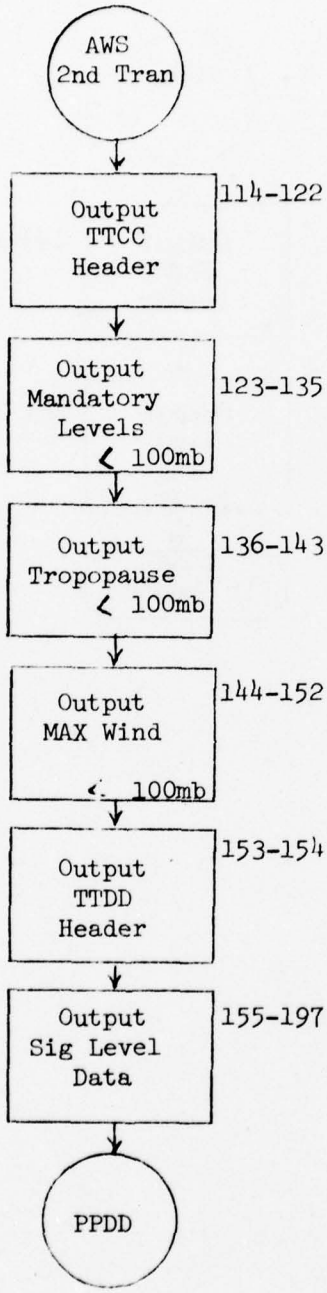


Figure 6j. Output AWS 2nd Transmission
(Track 0 File 16) (1 of 2)

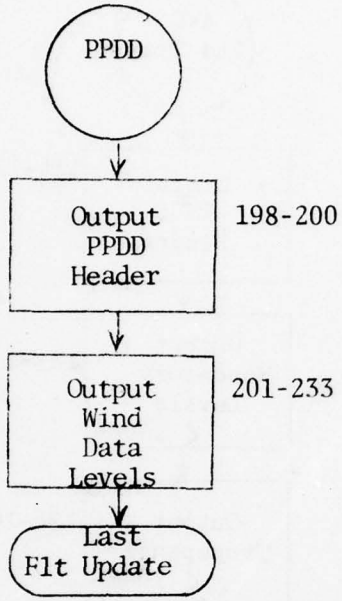


Figure 6k. AWS Output Continued (2 of 2)

e. Last Flight Update.

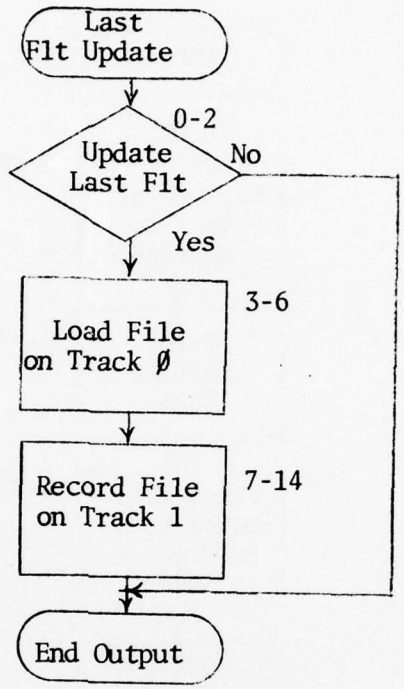


Figure 61. Last Flight Update

f. Flight Information.

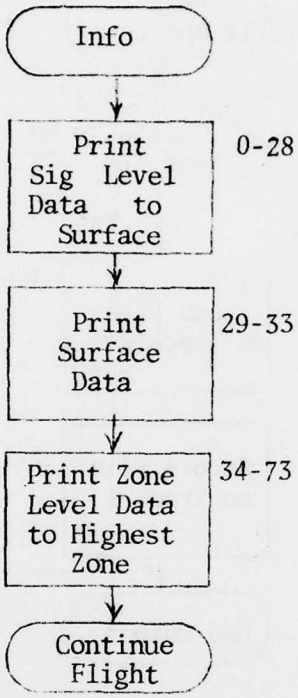


Figure 7. Flight Information (Track 1 File 9)

g. P. Tape.

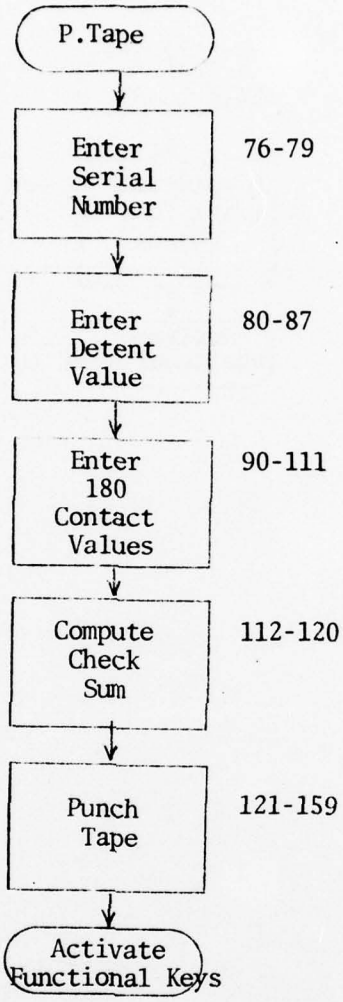


Figure 8. Pressure Tape (Track 1 File 2)

h. TACFIRE.

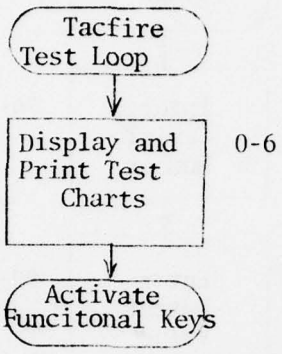


Figure 9. TACFIRE (Track 1 File 3)

5. SAMPLE TEST CASE.

Part a shows three flights conducted using the Meteorological Data Reduction Program.

Part b is the teletape output generated by the OUTPUT section. The UGC-74 teletape computer met message is in the correct format for the FADAC and the AWS messages are in the correct format for the Air Weather Service.

a. Test Flight

RADIOSONDE DATA
(FM 6-15)

Ft Sill

STATION	LOCATION		RELEASE TIME		DATE	HOUR	FLIGHT NO
MET	34.7	98.3		LST	1 march 1978	0900	1
				GMT	1 march 1978	1500	

LEVEL NUMBER	PRESSURE CONTACT (1)	PRESSURE MILLIBARS (2)	TEMPERATURE RECORDER DIVISION (3)	TEMPERATURE °C (4)	RELATIVE HUMIDITY RECORDER DIVISION (5)	RELATIVE HUMIDITY % (6)
BASE-LINE CHECK DATA	(INITIAL SETTING) 6.4	(INITIAL SETTING) 989	64.4	DRY 21.0 WET 11.8 DEP'N	81.0	
RELEASE DATA	6.0	989 REFERENCE		DRY WET DEP'N		
SUR 0	CONTACT	TIME	95.0	57.6	79.3	
1	10.6	1.70	95.0	52.4	65.6	
2	18.4	4.77	95.4	48.4	66.1	
3	21.0	5.75	95.0	48.0	71.4	
4	25.0	7.29	95.0	44.2	77.5	
5	29.3	8.87	94.8	48.4	79.8	
6	32.3	10.40	95.0*	XXX	XXX	
7	38.0	12.20	94.6*	39.0	75.8	
8	41.0	13.36	95.0	39.0	75.2	
9	46.0	15.27	95.0	34.3	46.1	
10						
11						
12						
13						
14						
15			QUADRANT	1		
16			ELEVATION	360 METERS		
17			OFF-SET	50 METERS		
18			SFC WIND DIR	110°		
19			SFC WIND SPD	04 KNOTS		
20						
21						
22						
23						
24						

(7) MODULATOR SERIAL NO.	BASELINE CHECK TIME	RECORDER OPERATOR	CHECKER
—	LST GMT		

43.00	00.0	212.85
44.00	00.1	213.85
46.95	00.2	247.80
47.45	00.3	263.60
44.10	00.4	280.40
41.20	00.5	283.15
40.45	00.6	285.45
39.65	00.7	285.40
39.10	00.8*	288.10
38.30	00.9	291.40
38.15	01.0	292.80
38.05	01.1	294.45
38.35	01.2	296.50
38.90	01.3	299.40
39.65	01.4	302.15
40.65	01.5	304.00
41.40	01.6	305.50
42.25	01.7	308.70
43.75	01.8	311.50
44.15	01.9	313.85
45.30	02.0	316.30
46.25	02.1	317.25
46.70	02.2	318.70
47.95	02.3	319.90
48.40	02.4	321.85
49.40	02.5*	323.15
49.55	02.6	324.90
50.30	02.7	327.40
51.50	02.8	329.80
51.85	02.9	331.05
51.25	03.0	333.90
51.25	03.1	335.80
50.30	03.2	337.75
50.70	03.3	340.15
50.55	03.4	341.85
50.20	03.5	343.15
50.45	03.6	345.35
50.75	03.7	348.00
50.90	03.8	350.95
51.85	03.9	351.45
51.15	04.0	354.00
51.60	04.1*	356.30
51.95	04.2	359.00
52.25	04.3	001.40
52.25	05.6	015.30
52.35	05.7	016.15
52.45	05.8	016.85
52.50	05.9*	017.55
52.50	06.0	017.40
52.30	06.1	018.15
52.30	06.2	017.75

51.40	07.4	28.90
51.30	07.5	28.80
51.20	07.6*	29.70
51.20	07.7	30.20
50.85	07.8	31.85
50.70	07.9	33.20
50.60	08.0	34.70
50.55	09.1	43.05
50.65	09.2	44.60
50.65	09.3*	45.80
50.40	09.4	46.50
50.40	09.5	47.05
50.40	09.6	48.25
50.40	09.7	48.95
50.30	09.8	49.80
50.30	09.9	50.20
50.45	10.0	50.70
49.85	11.0	58.25
49.70	11.1	59.70
49.60	11.2	59.80
49.55	11.3*	60.60
49.10	11.4	61.25
49.00	11.5	61.95
48.65	11.6	62.85
48.50	11.7	62.95
48.30	11.8	63.70
48.05	11.9	64.50
47.70	12.0	65.10
46.55	12.9	70.15
46.30	13.0	70.75
46.20	13.1*	71.65
46.10	13.2	72.30
46.10	13.3	73.05
46.10	13.4	73.75
46.10	13.5	74.65
42.80	14.8	88.70
42.55	14.9*	89.80
42.20	15.0	90.65
41.90	15.1	91.50
41.30	15.2	92.60
40.00	15.3	93.25

BAROSWITCH PRESSURE CALIBRATION CHART FLT # 1

BAROSWITCH SERIAL NO. 777-1092

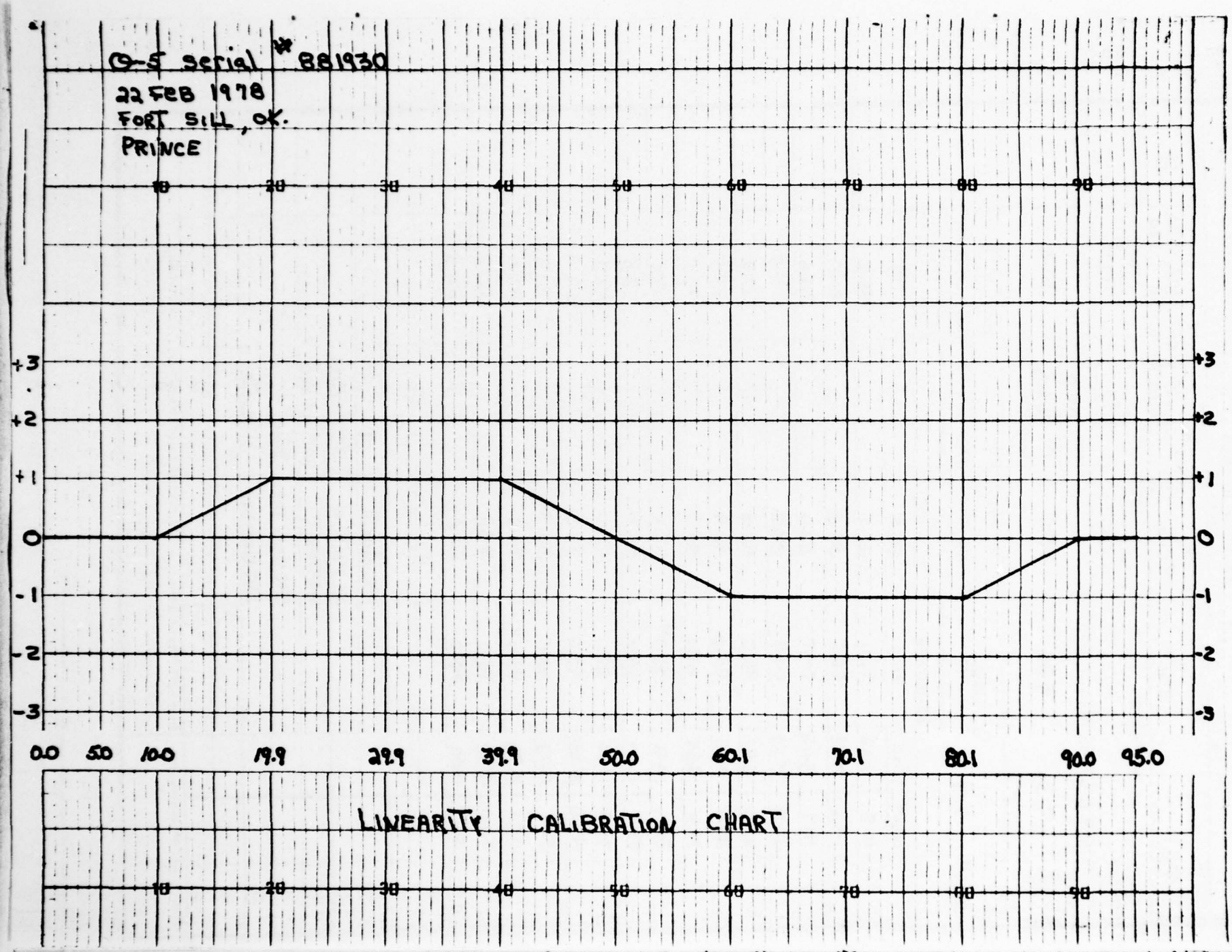
DETENT CLICK VALUE .60

	1	2	3	4	5	6	7	8	9
	1054.8	1042.8	1030.8	1019.3	1008.3	993.8	982.3	971.3	960.3
10	11	12	13	14	15	16	17	18	19
949.8	935.8	925.3	913.8	903.3	892.3	879.8	869.3	858.3	847.8
20	21	22	23	24	25	26	27	28	29
837.8	825.8	815.8	805.8	794.8	785.3	773.3	762.8	753.3	743.8
30	31	32	33	34	35	36	37	38	39
734.8	724.8	716.3	706.8	697.8	688.8	677.3	668.8	659.8	650.8
40	41	42	43	44	45	46	47	48	49
641.8	631.8	623.3	614.8	606.3	597.8	589.8	581.8	573.8	565.8
50	51	52	53	54	55	56	57	58	59
558.3	548.3	540.3	532.8	525.3	517.8	508.8	500.8	494.3	486.3
60	61	62	63	64	65	66	67	68	69
479.8	472.3	464.8	458.3	451.3	444.8	435.8	429.3	422.8	416.3
70	71	72	73	74	75	76	77	78	79
409.3	401.3	394.8	388.8	382.8	376.3	370.3	364.3	358.3	352.3
80	81	82	83	84	85	86	87	88	89
346.8	339.3	333.8	327.8	321.8	316.8	309.8	304.3	298.8	293.8
90	91	92	93	94	95	96	97	98	99
288.3	282.8	277.8	271.8	266.8	261.8	255.8	250.8	245.8	241.3
100	101	102	103	104	105	106	107	108	109
236.8	230.8	226.3	221.8	217.8	213.3	208.8	204.3	200.3	196.3
110	111	112	113	114	115	116	117	118	119
192.3	186.8	183.3	179.8	175.8	171.8	167.3	163.8	159.8	156.8
120	121	122	123	124	125	126	127	128	129
152.8	149.8	146.3	142.8	139.8	136.3	132.3	129.3	125.8	122.8
130	131	132	133	134	135	136	137	138	139
119.8	116.8	113.3	110.8	107.8	104.8	101.8	98.8	95.8	92.3
140	141	142	143	144	145	146	147	148	149
89.3	86.3	83.3	80.8	77.8	74.8	72.3	69.8	66.8	64.3
150	151	152	153	154	155	156	157	158	159
61.8	59.3	56.8	54.8	51.8	49.8	46.8	44.8	42.3	39.8
160	161	162	163	164	165	166	167	168	169
37.8	35.3	33.3	30.8	28.8	26.3	23.8	21.8	18.8	16.8
170	171	172	173	174	175	176	177	178	179
14.8	12.3	9.8	7.8	5.8	2.8	.0	.0	.0	.0

VIZ FORM NO 1292 068

VIZ MANUFACTURING CO.
335 E PRICE ST PHILA PA 19144 215 844 2626

0000 0000 0000 1000 61480 30084


O-5 serial # 881930
22 FEB 1978
FORT SILL, OK.
PRINCE
10
20
30
40
50
60
70
80
90
+3
+2
+1
0
-1
-2
-3
0.0
5.0
10.0
19.9
29.9
39.9
50.0
60.1
70.1
80.1
90.0
95.0
LINEARITY CALIBRATION CHART

RADIOSONDE DATA

(FM 6-15)

STATION	LOCATION		RELEASE TIME		DATE	HOUR	FLIGHT NO.
MET	34.7	98.3		LST	10 MARCH 1978	0730	2
				GMT	10 MARCH 1978	1330	

LEVEL NUMBER	PRESSURE CONTACT (1)	PRESSURE MILLIBARS (2)	TEMPERATURE RECORDER DIVISION (3)	TEMPERATURE °C (4)	RELATIVE HUMIDITY RECORDER DIVISION (5)	RELATIVE HUMIDITY % (6)
BASE-LINE CHECK DATA	(INITIAL SETTING) 7.3	(INITIAL SETTING) 980	75.0	DRY 39.0 WET 24.4 DEP'N	80.6	
RELEASE DATA	7.4	980		DRY WET DEP'N		
SUR 0	CONTACT	TIME	REFERENCE 95.0	60.5	50.5	
1	10.8	1.32	95.0	61.5	45.8	
2	14.0	2.48	96.5	55.8	41.5	
3	18.7	4.02	94.5	53.0	35.6	
4	22.0	5.57	94.7	50.3	75.0	
5	25.0	6.71	95.0	48.4	77.3	
6	29.0	8.28	95.0	48.4	78.5	
7	31.8	9.35	95.0	45.7	77.5	
8	33.4	10.2	95.0*	XX.X	XXX	
9	38.5	11.94	95.0*	42.6	73.3	
10	42.0	13.30	95.0	42.0	71.2	
11	47.0	15.23	95.0	38.3	62.2	
12	51.0	16.78	95.0	35.3	80.9	
13	53.6	17.80	95.0	34.6	82.0	
14	55.0	18.33	95.0	34.3	82.3	
? .4 15	56.5	19.00	95.0	33.7	82.3	
16						
17						
18				QUADRANT	1	
19				ELEVATION	360 METERS	
20				OFF-SET	100 METERS	
21				SFC WIND DIR.	320°	
22				SFC WIND SPD	11 KNOTS	
23						
24						

(7) MODULATOR SERIAL NO.	BASELINE CHECK TIME	RECORDER OPERATOR	CHECKER
	LST GMT		

00.85	00.0	176.65
23.85	00.1	188.00
36.30	00.2	192.55
38.55	00.3	196.50
39.40	00.4	195.20
40.00	00.5	197.55
	00.5*	
38.30	00.6	196.25
39.45	00.7	196.15
38.55	00.8	196.50
38.95	00.9	196.90
38.95	01.0	197.30
39.40	01.1	196.25
39.40	01.2	197.25
40.50	01.3	198.20
40.55	01.4	198.50
41.25	01.5	198.50
41.95	01.6	199.45
41.95	01.7	199.40
41.85	01.8	199.35
41.60	01.9	201.55
42.70	02.0	203.20
	02.0*	
43.60	02.1	204.30
44.50	02.2	204.50
44.25	02.3	204.90
45.40	02.4	205.40
46.30	02.5	207.35
47.20	02.6	210.15
47.35	02.7	211.70
48.40	02.8	213.50
49.65	02.9	213.85
50.30	03.0	214.40
51.55	03.1	214.25
52.30	03.2	214.55
52.85	03.3	215.10
53.05	03.4	215.30
	03.4*	
54.10	03.5	214.60
55.95	03.6	214.40
56.35	03.7	213.05
57.50	03.8	213.00

67.85	05.0	203.20
68.75	05.1	201.30
	05.1*	
70.15	05.2	200.45
72.55	05.3	198.30
72.60	05.4	198.25
72.95	05.5	198.05
74.50	05.6	193.65
73.95	05.7	190.50
74.10	05.8	188.15
74.65	05.9	184.35
74.45	06.0	181.50

73.45	06.7	154.45
73.60	06.8	150.50
73.95	06.9	147.80
	06.9*	
72.95	07.0	144.15
72.50	07.1	141.45
71.70	07.2	139.85
71.60	07.3	136.50
71.60	07.4	133.15
71.05	07.5	130.70
70.85	07.6	126.35
70.65	07.7	122.75
70.35	07.8	118.15

63.00	08.9	103.40
62.15	09.0	102.20
61.45	09.1	101.15
60.45	09.2	100.00
60.00	09.3	99.10
58.85	09.4	98.80
58.20	09.5	98.10
57.20	09.6	97.60
56.70	09.7	97.05
55.55	09.8	97.00
54.65	09.9	96.30
52.50	10.0	95.75

47.30	10.8	92.70
	10.8*	
46.60	10.9	92.40
46.15	11.0	92.05
45.30	11.1	91.90
44.45	11.2	91.95
43.90	11.3	92.00
43.25	11.4	92.35
42.75	11.5	92.55
42.80	11.6	92.75

39.15	12.5	92.50
37.80	12.6	92.30
37.50	12.7	92.10
37.20	12.8	91.80
36.85	12.9	91.60
	12.9*	
36.35	13.0	91.45
36.25	13.1	91.55
35.60	13.2	91.60

32.05	14.3	93.00
31.85	14.4	93.20
31.60	14.5	93.20
31.55	14.6	93.30
31.20	14.7	93.25
31.90	14.8	93.15
30.60	14.9	93.05
30.35	15.0	93.00
	15.0*	
30.35	15.1	92.90
29.80	15.2	92.90
29.50	15.3	92.80
29.40	15.4	92.75

28.60	15.8	92.35
28.45	15.9	92.35
28.25	16.0	92.40
28.00	16.1	92.40
17.70	16.2	92.40
27.75	16.3	92.40
27.70	16.4	92.40
27.35	16.5	92.20
27.15	16.6	92.10
27.00	16.7	91.90
26.85	16.8	91.85

25.10	17.8	91.30
24.90	17.9	91.20
24.80	18.0	91.10
24.65	18.1	91.05
24.50	18.2	91.00
24.20	18.3	90.95
24.15	18.4	90.85
24.00	18.5	90.80
23.90	18.6	90.80
23.75	18.7	90.70

Flight # 2

BAROSWITCH PRESSURE CALIBRATION CHART - FLT 2

BAROSWITCH SERIAL NO. 803-1000

DETENT CLICK VALUE .60

	1	2	3	4	5	6	7	8	9
	1057.8	1045.8	1033.8	1021.3	1009.8	995.3	983.8	971.8	960.8
10	11	12	13	14	15	16	17	18	19
949.8	935.3	923.8	912.8	902.3	891.3	877.8	867.8	856.8	846.8
20	21	22	23	24	25	26	27	28	29
835.8	823.8	813.3	802.8	792.8	782.8	770.8	760.8	750.8	741.3
30	31	32	33	34	35	36	37	38	39
731.8	721.8	712.3	702.8	693.8	684.8	673.3	664.3	655.3	646.8
40	41	42	43	44	45	46	47	48	49
637.8	627.3	618.8	609.8	601.8	592.8	584.8	576.8	568.8	560.8
50	51	52	53	54	55	56	57	58	59
552.8	542.8	534.8	527.3	519.8	512.3	502.8	495.3	488.3	480.8
60	61	62	63	64	65	66	67	68	69
473.8	466.8	459.8	452.8	445.8	438.8	430.3	423.8	416.8	410.8
70	71	72	73	74	75	76	77	78	79
403.8	396.3	389.8	383.8	377.3	371.3	365.3	359.3	353.3	347.3
80	81	82	83	84	85	86	87	88	89
341.8	334.8	328.8	322.8	317.3	311.8	305.3	299.8	294.8	289.3
90	91	92	93	94	95	96	97	98	99
284.3	279.3	274.3	269.3	264.3	259.8	253.8	248.8	244.3	239.8
100	101	102	103	104	105	106	107	108	109
235.3	229.8	225.3	220.8	216.8	212.8	208.3	204.3	199.8	196.3
110	111	112	113	114	115	116	117	118	119
191.8	187.8	183.8	179.8	175.8	172.8	167.8	164.3	160.8	157.3
120	121	122	123	124	125	126	127	128	129
153.8	150.8	147.3	143.8	140.8	137.8	133.8	130.8	127.3	124.3
130	131	132	133	134	135	136	137	138	139
121.3	117.8	114.8	111.8	108.8	106.3	102.8	99.8	96.3	92.8
140	141	142	143	144	145	146	147	148	149
89.8	86.8	83.8	80.8	77.8	74.8	71.8	68.8	65.8	62.8
150	151	152	153	154	155	156	157	158	159
59.8	56.8	54.3	51.3	48.8	45.8	42.8	39.8	37.3	34.3
160	161	162	163	164	165	166	167	168	169
31.8	28.8	25.8	22.8	20.3	17.3	14.8	11.8	8.8	5.8
170	171	172	173	174	175	176	177	178	179
2.8	.0	.0	.0	.0	.0	.0	.0	.0	.0

VIZ FORM NO. 1292-068

VIZ MANUFACTURING CO.
335 E. PRICE ST. PHILA. PA. 19144 215 844-2020

0000 0000 0000 0010 56545 29084

RADIOSONDE DATA

(FM 6-15)

STATION	LOCATION		RELEASE TIME		DATE	HOUR	FLIGHT NO
MET	34.7	98.3		LST	8 Aug 1978	1500	3
				GMT	8 Aug 1978	2100	

LEVEL NUMBER	PRESSURE: CONTACT (1)	PRESSURE: MILLIBARS (2)	TEMPERATURE: RECORDER DIVISION (3)	TEMPERATURE: °C (4)	RELATIVE HUMIDITY: RECORDER DIVISION (5)	RELATIVE HUMIDITY: % (6)
BASE-LINE CHECK DATA	(INITIAL SETTING) 8.2	(INITIAL SETTING) 968.6	71.5	DRY 34.5 WET 23.6 DEP'N	80.4	
RELEASE DATA	8.2	968.6 REFERENCE		DRY WET DEP'N		
SUR 0	CONTACT	TIME	95.0	71.4	78.5	
1	13.0	1.70	95.0	68.7	74.6	
2	18.7	3.47	95.0	65.7	69.3	
3	19.9	4.29	94.9	67.1	77.3	
4	26.0	6.75	95.0	64.8	80.1	
5	31.0	8.70	94.8	62.1	79.3	
6	33.6	10.01	94.6	59.5	74.2	
7	37.0	11.27	95.0	58.2	72.3	
8	40.0	12.43	94.9	57.2	82.4	
9	48.0	16.01	94.7	51.6	78.3	
10	51.0	17.51	95.0	49.9	82.8	
11	57.1	20.47	95.0	45.3	83.7	
12	60.0	21.54	95.0	44.0	83.7	
13	63.0	22.83	95.0	42.8	83.8	
14	71.2	26.71	95.0	36.9	84.2	
15	78.0	30.11	95.0	31.2	84.5	
16	81.0	31.59	95.0	28.7	84.3	
17	87.7	34.43	95.0	24.6	85.3	
18	95.5	38.63	95.0	18.1	XXX	
19	97.1	39.81	95.1	16.1	XXX	
20				QUADRANT #	1	
21				ELEVATION	400 meters	
22				OFFSET	44 meters	
23				Sfc wind DIR.	331°	
24				Sfc wind SPD.	08 knots	

(7) MODULATOR SERIAL NO.	BASELINE CHECK TIME	RECORDER OPERATOR	CHECKER
	LST GMT		

USE FLT # 1 PRESSURE CALIBRATION CHART

AD-A066 218 ARMY ELECTRONICS RESEARCH AND DEVELOPMENT COMMAND FO--ETC F/G 4/2
DOCUMENTATION OF SOFTWARE IN THE OL-192 METEOROLOGICAL DATA RED--ETC(U)
FEB 79 R BELLUCCI, S W BURNETT, T RICHTER

UNCLASSIFIED DELCS-TR-79-1 NL

2 OF 2
AD
A066218

END
DATE
FILMED
5-79
DDC

00.85	00.0	176.65
23.85	00.1	188.00
36.30	00.2	192.55
38.55	00.3	196.50
39.40	00.4	195.20
40.00	00.5	197.55
	00.5*	
38.30	00.6	196.25
39.45	00.7	196.15
38.55	00.8	196.50
38.95	00.9	196.90
38.95	01.0	197.30
39.40	01.1	196.25
39.40	01.2	197.25
40.50	01.3	198.20
40.55	01.4	198.50
41.25	01.5	198.50
41.95	01.6	199.45
41.95	01.7	199.40
41.85	01.8	199.35
41.60	01.9	201.55
42.70	02.0	203.20
	02.0*	
43.60	02.1	204.30
44.50	02.2	204.50
44.25	02.3	204.90
45.40	02.4	205.40
46.30	02.5	207.35
47.20	02.6	210.15
47.35	02.7	211.70
48.40	02.8	213.50
49.65	02.9	213.85
50.30	03.0	214.40
51.55	03.1	214.25
52.30	03.2	214.55
52.85	03.3	215.10
53.05	03.4	215.30
	03.4*	
64.90	04.8	206.10
66.40	04.9	203.20
67.85	05.0	203.20
68.75	05.1	201.30
	05.1*	
70.15	05.2	200.45
72.55	05.3	198.30
74.15	06.5	161.80
73.70	06.6	153.75
73.45	06.7	154.45
73.60	06.8	150.50
73.95	06.9	147.80
	06.9*	

67.75	08.3	108.80
67.15	08.4	107.55
66.30	08.5	105.50
65.60	08.6	105.65
64.95	08.7	105.10
63.75	08.8	103.90
	08.8*	
50.70	10.3	94.75
50.45	10.4	94.20
49.25	10.5	93.80
48.55	10.6	93.35
48.00	10.7	93.00
47.30	10.8	92.70
	10.8*	
41.80	11.9	93.20
40.55	12.0	92.40
40.05	12.1	93.45
39.60	12.2	93.25
39.20	12.3	93.00
38.70	12.4	92.75
33.60	13.8	92.45
33.35	13.9	92.60
33.10	14.0	92.75
32.75	14.1	92.85
32.40	14.2	92.90
32.05	14.3	93.00
28.60	15.8	92.35
28.45	15.9	92.35
28.25	16.0	92.40
28.00	16.1	92.40
17.70	16.2	92.40
27.75	16.3	92.40
27.70	16.4	92.40
24.90	17.9	91.20
24.80	18.0	91.10
24.65	18.1	91.05
24.50	18.2	91.00
24.20	18.3	90.95
24.15	18.4	90.85
24.00	18.5	90.80

20.15	21.6	89.55
20.05	21.7	89.45
20.00	21.8	89.45
19.85	21.9	89.35
19.70	22.0	89.25
19.65	22.1	89.25
19.55	22.2	89.15
19.45	22.3	89.00
16.55	25.3	88.10
16.40	25.4	88.10
16.45	25.5	88.10
16.25	25.6	88.10
16.15	25.7	88.05
16.05	25.8	88.10
13.30	29.4	88.45
13.20	29.5	88.50
13.15	29.6	88.55
13.10	29.7	88.50
13.05	29.8	88.55
13.05	29.9	88.55
13.00	30.0	88.55
	32.8 *	
11.30	32.9	87.65
11.30	33.0	87.60
11.25	33.1	87.55
11.20	33.2	87.55
11.10	33.3	87.55
11.10	33.4	87.50
11.00	33.5	87.45
10.25	36.7	87.20
10.25	36.8	87.20
10.20	36.9	87.20
10.20	37.0	87.20
10.10	37.1	87.20
10.05	37.2	87.15
10.00	37.3	87.15
10.00	37.4	87.15
10.00	37.5	87.15
	37.5 *	

Flight #1

b. Punch Output.

```
METSR1347983
081440+07023208
9

METCM1347983
081440036989
00196004285350989
01211010281900977
02272009277800948
03375008273900902
04416010270600847
05425009267107795
06477013268807746
07501013265307700
08500018259406656
09555028256306614
9

METFM1347983
081440036TR0
00196004
01373007
02518017
9

METB31347983
081440036976
00200498499 2
01211098199 5
02260997400 0
03340797000 3
04380896800 4
05400896700 4
06461097699 8
07511597399 7
9
```

```
METB21347983
08144C036976
20200498499 2
012110981995
022409977998
032907974001
043307972003
053607971003
064107971002
074508972000
9


METH1 34714 98327 08036
 TTAA 58147 RB123 99989 09864 11004 00268 ///// ///// 85583
03357 23510 70/// ///// ///// 88999;
 TTBB 5814/ RB123 00989 09864 11935 02458 22850 03357 33821
03559 44780 08561 55737 02963 66700 03361 77655 14760 88623
14959 99586 21147;
 PPBB 58140 RB123 90023 11004 13511 20005 90467 21510 23510
24010 9089/ 26012 27013 9123/ 28519 30525;
```

Flight #2

```
METSP1347933
081430+12702507
9

METCM1347983
08143003698 0
00569310285109 80
01051003728560968
02041009285609 40
03128005277008 95
04426005040908 41
05483011268107 90
06456017266407 41
07471027262506 95
08482031260006 51
09493035257906 10
10479037253705 70
11473043250205 33
9

METFM1347983
081480036TP0
00569011
01545003
02479027
9

METB31347983
08143003 6967
00571198997 8
01058799397 4
02040999797 0
03100598497 9
04540198098 3
05490697698 5
06471597598 6
07482297593 5
08482897498 4
9
```

```
METB21347983
081480036967
005711989978
010507993974
020508995972
030706991975
040704988979
050102985981
064905981984
074811979985
084815978985
9

METV1 34714 98350 08036
 TTAA 58157 RB123 99980 10856 32011 00191 ///// ///// 85528
00239 28004 70/// ///// ///// 50557 24765 88999 77999;
 TTBB 5815/ RB123 00980 10856 11940 12449 22903 02644 33850
00239 44784 06361 55742 06362 66715 09761 77619 14358 88544
22763 99513 23965 11500 24765;
 PPBB 58150 RB123 90023 32011 02509 06506 90467 06003 25508
27011 9089/ 26015 26019 91246 27031 27535 27040;
```

Flight #3

```
METSP1347983
081530+34302709
9

METCN1347983
081530040969
0058300831040969
0103601130930958
0203901030650932
0310800730230891
0443000429970842
0547401029730795
0645101529400751
0747202528880708
0848503223470667
0943303428140628
1048203627710591
1147403927300556
1247304826710505
1347106426100444
1447908025340389
1547009824500339
1647208223760295
9

METFN1347983
081530040TP0
00588008
01589003
02476026
03475043
04476072
05471090
9

METB31347983
081530040956
0059080778888
010411076889
020410073892
030307069896
040202070898
055005273899
064713072903
074822069908
084326067913
094032063918
104852063926
114761063937
9
```

```
METB21347983
08153004095 6
00590807788 8
01041107688 9
02041007489 1
03060907289 4
04060607189 6
05030407189 7
06520507290 0
07491007190 3
08491407190 5
09481807090 9
10482306891 5
11483706792 2
9
```

```
METV1 34715 98319 08040
 TTAA 58153 PB123 99969 34264 33008 00109 ///// ///// 85563
23058 20505 70243 12860 26526 50597 07366 26550 40768 17966
27077 30975 34367 26584 25100 473// 83999 77321 26598 418//;
 TTBB 5815/ PB123 00969 34264 11850 23058 22838 25662 33724
17063 44700 12860 55668 10459 66641 09066 77573 01261 88500
07366 99458 10566 11300 34367 22250 473//;
 PPBB 58150 PB123 90023 33008 02011 03012 90467 08007 25006
26509 9089/ 26013 25517 91246 27532 27034 27038 9205/ 26551
27076 9302/ 26594 26584;
```

6. PROGRAM LISTING.

a. System Listing

(1) Master Supervisory File

TRACK 0 FILE 0

```
0: prt "OL-192 ARTILLERYMETEOROLOGY....."
1: prt "DATED:17 JULY 78";spc 2
2: prt "TAPE NO.    M0000"
3: dim A[11]
4: trk 0;ldf 4,A[*]
5: fxd 0;prt "COUNTER",A[11];spc 2
6: trk 0;ldk 1
7: dsp "USE FUNCTIONAL KEYS TO ACTIVATE";gsb "BBB"
8: stp
9: dsp "PLEASE USE UPPER ROW KEYS";gsb "BBB"
10: stp
11: dsp "LET'S TRY IT AGAIN";gsb "BBB"
12: gto -6
13: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
14: " THIS PROGRAM REQUIRES THE FOLLOWING HARDWARE":
15: "      HP 9825A CALCULATOR WITH 24K RWM":
16: "            HP9825A OPERATING SYSTEM ROM 67905":
17: "            GENERAL IO/ EXTENDED IO ROM 98213A":
18: "            STRING-ADV PROGRAMMING ROM 98210A":
19: "            REMEX READER/PERFORATOR FOR PUNCHED OUTPUT":
20: "            INTERFACE CABLE 98032A OPT  A03":
21: end
*762
```

(2) Functional Keys

TRACK 0 FILE 1

```
f0=f12:   *dsp "PROCESSING START OF FLIGHT"; trk0; ldp 2
f1-f13:   *dsp "PROCESSING FLIGHT CONTINUATION"; trk0; ldp6
f2=f14:   *dsp "PROCESSING LAST FLIGHT MESSAGES"; trk1; ldp 4
f3=f15:   *dsp "PROCESSING TACFIRE TEST LOOP"; trk1; ldp 3
f4=f16:   *dsp "PROCESSING PRESSURE CALIBRATION"; trk1; ldp 2
f5=f17:   *dsp "PROCESSING PIBAL"; trk1; ldp 1
f6=f18:   *dsp "PROCESSING FLIGHT INFORMATION"; trk1; ldp 9
f7=f19:   *dsp "PROCESSING MET MESSAGES"; trk0; ldp 12
f8=f20:   *gsb "DELETE"
f9=f21:   /-2tn↑10
f10=f22:  /-1tn↑10
f11=f23:  /
```

b. Preflight

(1) Date, Survey, Linearity, Pressure

TRACK 0 FILE 2

```
0: dim A$[100],L$[16],B$[100],R[15],C$[32]
1: dim W$[56],O$[895],V$[96],B[25]
2: dim P$[102],Q$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
3: trk 0;ldf 9,P$,Q$,O$,L$,N$,G$,D$
4: "                    "→D$[1,16]
5: "ENTER YEAR (1900 to 1999)"→C$;gsb "DI"
6: if L<1900 or L>1999;gto -1
7: val(A$[3,4])→R[1];A$[3,4]→D$[14,15]
8: "ENTER NUMBER OF MONTH (1 to 12)"→C$;gsb "DI"
9: L→R[2];if R[2]<1 or R[2]>12;gto -1
10: "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"→A$
11: A$[3*R[2]-2,3*R[2]]→D$[10,12]
12: 31→N;if R[2]=2;28→N;if int(R[1]/4)=R[1]/4;29→N
13: if R[2]=4 or R[2]=6 or R[2]=9 or R[2]=11;30→N
14: str(N)→A$;A$[2]→A$[1]
15: "ENTER DAY OF MONTH (1 to "→C$[1,25];A$[1,2]→C$[26,27]
16: ")"→C$[28];gsb "DI"
17: L→R[3];if R[3]<1 or R[3]>N;gto -3
18: if R[3]>9;str(R[3])→D$[1,3];gto +2
19: str(R[3])→D$[2,3];"0"→D$[2,2]
20: D$[2,3]→D$[1,2];" "→D$[3,3]
21: 2→A;"0"→A$;beep;wait 150;beep
22: ent "ENTER ZULU TIME (0000 to 2359)",A$
23: if A≠2 or len(A$)≠4;gto -2
24: val(A$)→R[4];if R[4]<0 or R[4]>2359;gto -3
25: if frc(R[4]/100)>.59;gto -4
26: A$[1,4]→D$[4,7];"Z"→D$[8,8]
27: "IS THE DATE "→C$[1,12];D$→C$[13,29];"?"→C$[30];gsb "Y/N"
28: if A=-2tn^10;gto -24
29: dsp D$[1,16];gsb "BBB"
30: prt "DATE:",D$;spc 2
31: D$[1,2]→P$[13,14];D$[4,5]→P$[15,16]
32: val(D$[6,7])→A→B→C;prnd(A/6,0)→A;int(B/6)→B;if A>9;"9"→P$[17,17];gto +3
33: if Bmod2=0 and Cmod6=3;A-1→A
34: str(A)→A$;A$[2,2]→P$[17,17]
35: "IS SURVEY OF "→C$[1,13];O$→C$[14,29];"OK?"→C$[30];gsb "Y/N"
36: if A=-1tn^10;gto +48
37: dsp "PERFORM SURVEY CONTROL";gsb "BBB"
38: D$[1,16]→O$[1,16]
39: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
40: ent "ENTER STATION ELEVATION (meters)",A$
41: if A≠2;gto -2
42: if val(A$)<-200 or val(A$)>5000;gto -3
43: int(val(A$[1,4])/10+.5)→K;fxd 0
44: if K>=100;str(K)→P$[18,21];"0"→P$[18,18];gto +6
45: if K>=10;str(K)→P$[19,21];"00"→P$[18,19];gto +5
46: if K>=0;str(K)→P$[20,21];"000"→P$[18,20];gto +4
47: int(abs(val(A$[1,4])/10-.5))→K;"0-"→P$[18,19]
48: if K>=10;str(K)→P$[19,21];"0-"→P$[18,19];gto +2
49: if K>=0;str(K)→P$[20,21];"0-0"→P$[18,20]
*23621
```

```
50: "IS THE STATION ELEV "→C$[1,20];A$[1,4]→C$[21,24];" METERS?"→C$[25]
51: gsb "Y/N"
52: if A=-2tr^10;gto -13
53: "IS THIS A COBLE LOCATION ?"→C$;gsb "Y/N"
54: if A=-2tr^10;gto +7
55: 2→A;""→A$;"0"→A$;beep;wait 150;beep
56: cnt "ENTER CODE(6 characters)",A$
57: if A#2 or len(A$)#6;gto -2
58: "IS CODE LOCATION "→C$[1,18];A$[1,6]→C$[19,25];"?"→C$[26];gsb "Y/N"
59: if A=-2tr^10;gto -4
60: car(A$[1,6])→B$[7,12];"9"→B$[6,6];gto +24
61: "ENTER LATITUDE (0.0 to 90.0)"→C$;gsb "LI"
62: if L<0 or L>90;gto -1
63: L→K;str(K)→A$[1,5]
64: "IS THE LATITUDE "→C$[1,16];A$[2,5]→C$[17,20];" DEGREES ?"→C$[21]
65: gsb "Y/N"
66: if A=-2tr^10;gto -5
67: A$[2,3]→B$[7,8];A$[5,5]→B$[9,9]
68: if K<10;"0"→B$[7,7];A$[2,2]→B$[8,8];A$[4,4]→B$[9,9]
69: "ENTER LONGITUDE (0.0 to 99.9)"→C$;gsb "LI"
70: if L<0 or L>99.9;gto -1
71: L→K;str(K)→A$[1,5]
72: "IS THE LONGITUDE"→C$[1,16];A$[2,5]→C$[18,21];"DEGREES ?"→C$[23]
73: gsb "Y/N"
74: if A=-2tr^10;gto -5
75: A$[2,3]→B$[10,11];A$[5,5]→B$[12,12]
76: if K<10;"0"→B$[10,10];A$[2,2]→B$[11,11];A$[4,4]→B$[12,12]
77: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
78: cnt "ENTER OCTANT (0 to 8 not 4)",A$
79: if A#2 or len(A$)#1;gto -2
80: if val(A$)<0 or val(A$)=4 or val(A$)>8;gto -3
81: "IS OCTANT "→C$[1,12];A$[1,1]→C$[13,13];" ?"→C$[14];gsb "Y/N"
82: if A=-2tr^10;gto -5
83: A$[1,1]→B$[6,6]
84: dsp "SURVEY DATA";gsb "BLB"
85: prt "SURVEY DATED:",C$
86: if val(B$[6,6])=9;prt "COBED LOCATION:",B$[7,12];gto +4
87: fxd 0;prt "OCTANT:",val(B$[6,6])
88: fxd 1;prt "LATITUDE:",val(B$[7,9])/10
89: fxd 1;prt "LONGITUDE:",val(B$[10,12])/10
90: fxd 0;val(B$[19,21])*10→K;prt "ELEVATION:",K
91: B$[1,12]→B$[25,36]→B$[37,48]→B$[49,60];"B3"→B$[28,29];"B2"→B$[40,41]
92: "BT"→B$[52,53];"CM"→B$[4,5];B$[13,24]→B$[61,72];"TIC"→B$[70,72]
93: B$[1,12]→B$[73,84];"SB"→B$[76,77]
94: B$[13,24]→B$[85,96];"+TTTDDDFF"→B$[91,99]
95: spc 2;"IS A TTY-76 TELETALK RECORD ?"→C$[1,32];gsb "Y/N"
96: if A=-1tr^10;gto +3
97: dsp "REMEX USES ONE-INCH WIDE TAPE";gsb "BLB"
98: prt "OUTPUT:UCC-74";"ONO"→B$[100,102];gto +3
99: dsp "REMEX USES 11/16 INCH WIDE TAPE";gsb "BLB"
*29112
```

```
100: prt "OUTPUT:TTY-76";"0YC"→L$[100,102]
101: spc 3;"IS ECD CK OF"→C$[1,12];L$[1,15]→C$[14,28];"OK?"→C$[30]
102: gsb "Y/N"
103: if A=-2tn^10;gto +4
104: "IS ECD SN"→C$[1,9];N$[1,6]→C$[11,16];"BEING USED ?"→C$[18]
105: gsb "Y/N"
106: if A=-1tn^10;gto +41
107: dsp "PERFORM RECORDER LINEARITY CHECK";gsb "EEL"
108: L$[1,16]→L$[1,16]
109: ""→A$;"0"→A$[1,1];2→A;beep;wait 150;beep
110: prt "ENTER ECD SERIAL NUMBER (xxxxxx)",A$
111: if A#2 or len(A$)#6;gto -2
112: "IS SERIAL NUMBER"→C$[1,16];A$[1,6]→C$[18,23];"?"→C$[25]
113: gsb "Y/N"
114: if A=-2tn^10;gto -5
115: A$[1,6]→L$[1,6]
116: fxd 0;190→K;str(K)→E$[1,4];E$[2,4]→E$[1,3];gsb "LINEARITY"
117: E$[1,16]→C$[1,16]
118: 0→I
119: for K=180 to 20 by -20;str(K)→E$[1,4];E$[2,4]→E$[1,3];gsb "LINEARIT
120: E$[1,16]→C$[16I+1,16I+16];next K
121: for K=10 to 0 by -10;str(K)→E$[1,4];E$[2,4]→E$[1,3];gsb "LINEARITY"
122: E$[1,16]→C$[16I+1,16I+16];next K
123: gto +24
124: "LINEARITY":
125: if K<100;" "→E$[3,3];if K<10;"  "→E$[2,3]
126: 2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
127: fxd 0;dsp "ENTER ECD ORDINATE AT",E$[1,3],"EEPTZ";ent "",A$[2]
128: if A#2;gto -2
129: "0000000000000000"→E$[1,16]
130: if val(A$)>95.3;"SIC ISN'T  RIGHT"→E$[1,16];gto +15
131: if A$[2,2]="-";"S.C. ISN'T RIGHT"→E$[1,16];gto +14
132: if len(A$)>5;gto -8
133: str(K)→E$[1,4];E$[2,4]→E$[1,3]
134: if K<100;E$[1,2]→E$[2,3];" "→E$[1,1]
135: if val(E$[1,3])=0;"00"→E$[2,3]
136: str(K/2)→E$[4,6]
137: if K=0;" 00"→E$[4,6]
138: if K=10;" 05"→E$[4,6]
139: fxd 1;str(val(A$))→E$[7,11]
140: if val(A$)<10;str(val(A$))→E$[8,11]
141: if val(A$)=0;"0.0"→E$[9,11]
142: str(val(E$[5,6])-val(E$[8,11]))→E$[12,16]
143: if val(E$[13,16])<10;E$[13,15]→E$[14,16];" "→E$[13,13]
144: if abs(val(E$[13,16]))>.3;"*"→E$[12,12]
145: I+1→I
146: ret
147: dsp "AN/TIQ-5 RECORDER LINEARITY DATA";gsb "EEL"
148: prt "RECORDER S/N:",N$,"CALIBRATION ON:",E$[1,16];spc 1
149: prt "          A       ","  I       C    E ","  N   T    T    E "
*8514
```

```
150: prt " P  P    U    R ","U  U    A    C ","T  E    I    P ";spc 1
151: prt " Pz PD   RD   RD ",C$;spc 1
152: 0→K
153: for I=1 to 10
154: if C$[12+16I,12+16I]="*" or C$[16I,16I]="1";-5001→K
155: next I
156: if K#-5001;spc 1;gto +5
157: dsp "REPAIR RECORDER,THEN TRY AGAIN";gsb "ERR"
158: prt "OUT OF TOLERANCEVALUES INDICATED BY * (*)";spc 1
159: prt "OUT OF TOLERANCEVALUES MAY BE  EITHER (+) OR(-)"
160: spc 1;prt "RECALIBRATE RCD";spc 1;gto -44
161: "IS AN/ART-4 SONDE BEING USED ?"→C$;gsb "Y/N"
162: if A=-2tn^10;gto +4
163: "4"→P$[102,102];dsp "USING ARMY RADIOSONDE AN/ART-4";gsb "ERR"
164: prt "RADIOSONDE TYPE:   ARMY AN/ART-4";spc 2
165: gto +3
166: "1"→P$[102,102];dsp "USING NWS RADIOSONDE JCO5 SERIES";gsb "ERR"
167: prt "RADIOSONDE TYPE: NWS JCO5 SERIES";spc 2
168: gto +54
169: "D1":fxd 1;2→A;""→A$;"0"→A$[1,1];Beep;wait 150;Beep;dsp C$[1,32]
170: ent "",A$;if A$[1,1]="0" or A$[1,1]="1" or A$[1,1]="2";gto +7
171: if A$[1,1]="3" or A$[1,1]="4" or A$[1,1]="5" or A$[1,1]="6";gto +6
172: if A$[1,1]="7" or A$[1,1]="8" or A$[1,1]="9" or A$[1,2]=".0";gto +5
173: if A$[1,2]=".1" or A$[1,2]=".2" or A$[1,2]=".3" or A$[1,2]=".4";gto
174: if A$[1,2]=".5" or A$[1,2]=".6" or A$[1,2]=".7" or A$[1,2]=".8";gto
175: if A$[1,2]=".9";gto +2
176: gto -7
177: if A#2;gto -8
178: if A$[3,3]="*" or A$[4,4]="*" or A$[5,5]="*" or A$[6,6]="*";sfg 11
179: val(A$)→L;str(L)→A$;A$[2]→A$[1];if L<0 or L>1999;gto -10
180: "                                  "→C$[1,32]
181: ret
182: "Y/N":2→A;Beep;dsp C$[1,32];ent "",A
183: if A#-2tn^10 and A#-1tn^10;gto -1
184: "                                  "→C$[1,32]
185: ret
186: "DELETE":dsp "ACTIVATED WITH FIRST SIG LEVEL";gsb "ERR"
187: ret
188: "PRESS CONF":if R[2]>=1;0→J;4→I;0→Y
189: if R[2]>5;I→J;9→I;1→Y
190: if R[2]>10;I→J;14→I;2→Y
191: if R[2]>15;I→J;19→I;3→Y
192: if R[2]>20;I→J;24→I;4→Y
193: if R[2]>25;I→J;29→I;5→Y
194: if R[2]>30;I→J;39→I;6→Y
195: if R[2]>40;I→J;49→I;7→Y
196: if R[2]>50;I→J;59→I;8→Y
197: if R[2]>60;I→J;79→I;9→Y
198: if R[2]>80;I→J;99→I;10→Y
199: if R[2]>100;I→J;119→I;11→Y
*18346
```

```
200: if R[2]>120;I→J;139→I;12→Y
201: if R[2]>140;I→J;R[9]-1→I;13→Y
202: val(U$[5J+1,5J+5])→R[4];val(U$[5I+1,5I+5])→R[3];int(R[2])→A
203: frc(R[2])→J
204: (val(U$[5A-4,5A])-val(U$[5A+1,5A+5]))(-J)+val(U$[5A-4,5A])→R[5]
205: val(V$[6Y+1,6Y+6])→R[6];val(V$[6Y+7,6Y+12])→R[7]
206: R[6]-(R[6]-R[7])(R[4]-R[5])/(R[4]-R[3])→R[8]
207: rct
208: "PRESS CHECK":if R[2]=5;0→J;9→I;0→Y
209: if R[2]=15;5→J;19→I;2→Y
210: if R[2]=25;19→J;29→I;4→Y
211: if R[2]=40;29→J;49→I;6→Y
212: if R[2]=60;49→J;79→I;8→Y
213: if R[2]=100;79→J;119→I;10→Y
214: if R[2]=120;99→J;139→I;11→Y
215: val(U$[5J+1,5J+5])→R[4];val(U$[5I+1,5I+5])→R[3];R[2]→A
216: val(U$[5A-4,5A])→R[5]
217: val(V$[6Y+1,6Y+6])→R[6];val(V$[6Y+13,6Y+18])→R[7]
218: R[6]-(R[6]-R[7])(R[4]-R[5])/(R[4]-R[3])→R[8]
219: val(V$[6Y+7,6Y+12])→A;if A+1+.005A>R[8] and A-1-.005A<R[8];gto +2
220: -9→W
221: rct
222: "DO YOU HAVE A PRESSURE TAPE ?   "→C$[1,32];gsb "Y/N"
223: if A=-1tr^10;gto +93
224: dsp "PERFORM PAROSWITCH CALIBRATION";gsb "BEE"
225: "10673105531038810278101681000309888097730965809548"→U$[1,50]
226: "09403092880917309063089580802308713086080849808398"→U$[51,100]
227: "08268081630806307963078630773307638075380743807343"→U$[101,150]
228: "07238071530705306958068680675806663065680648306393"→U$[151,200]
229: "06283061980610806023059380585305768056880560805528"→U$[201,250]
230: "05428053430526805188051130501804943048680479304723"→U$[251,300]
231: "04643045730450304433043630427804208041380407304008"→U$[301,350]
232: "03923038580379303728036680360803543034830342303363"→U$[351,400]
233: "03288032280317303113030580299302938028830282802773"→U$[401,450]
234: "02723026730261802568025230245802408023630231302268"→U$[451,500]
235: "02208021680212302073020330198801948019030186301823"→U$[501,550]
236: "01773017330169301653016180156801533014980146301428"→U$[551,600]
237: "01388013530132301293012580121801188011580112801098"→U$[601,650]
238: "01063010330100300978009480091800883008530082300793"→U$[651,700]
239: "00783007730076300753007430073300723007130070300693"→U$[701,750]
240: "00683006730066300653006430063300623006130060300593"→U$[751,800]
241: "00583005730056300553005430053300523005130050300493"→U$[801,850]
242: "00483004730046300453004430043300423004130040300403"→U$[851,895]
243: 1→I;0→J;gsb "PRESSURE"
244: for I=5 to 30 by 5;gsb "PRESSURE"
245: next I
246: for I=40 to 60 by 10;gsb "PRESSURE"
247: next I
248: for I=80 to 140 by 20;gsb "PRESSURE"
249: next I
*32418
```

```
250: gto +12
251: "PRESSURE":fxd 0;str(I)→I$
252: "ENTER PRESSURE AT CONTACT NO"→C$[1,28];I$[1,4]→C$[29,32];gsb "DI"
253: if A#2;gto -1
254: if val(A$)<.1 or val(A$)>1100;gto -2
255: if len(A$)=5;A$[1,5]→A$[2,6];"0"→A$[1,1]
256: if len(A$)=4;A$[1,4]→A$[3,6];"00"→A$[1,2]
257: if len(A$)=3;A$[1,3]→A$[4,6];"000"→A$[1,3]
258: if len(A$)=2;A$[1,2]→A$[5,6];"0000"→A$[1,4]
259: A$[1,6]→V$[6J+1,6J+6]
260: J+1→J
261: ret
262: "ENTER CONTACT NO OF LOWEST PRESS"→C$[1,32];gsb "LI"
263: fxd 0;if L<141 or L>179;gto -1
264: L→I;gsb "PRESSURE"
265: val(B$)→I→R[9]→B[14]
266: dsp "LAROSWITCH PRESSURE DATA";gsb "EIL"
267: prt "LAROSWITCH PRESSCALIBRATION      ";spc 1
268: prt "---------------","        1        5"
269: "   "→A$[1,2];V$[1,6]→A$[3,8];"   "→A$[9,10];V$[7,12]→A$[11,16]
270: if val(V$[1,6])<1000;" "→A$[3,3]
271: if val(V$[7,12])<1000;" "→A$[11,11]
272: prt A$[1,16]
273: prt "   10          15"
274: V$[13,18]→A$[1,6]
275: if val(A$[1,6])<1000;A$[2,6]→A$[1,5];" "→A$[6,6]
276: "     "→A$[7,10];V$[19,24]→A$[11,16]
277: if val(A$[11,16])<1000;" "→A$[11,11]
278: prt A$[1,16]
279: prt "   20          25"
280: V$[26,30]→A$[1,5];"       "→A$[6,11];V$[32,36]→A$[12,16]
281: prt A$[1,16]
282: fmt 4,2x,f3.0,11x
283: for N=0 to 3;wrt 16.4,30+10N;prt V$[38+6N,42+6N];next N
284: if val(V$[80,84])<100;" "→V$[80,80]
285: for N=0 to 3;spc 2;wrt 16.4,80+20N;prt V$[62+6N,66+6N];next N
286: fxd 0;prt R[9];fxd 1;prt val(V$[85,90])
287: if val(V$[80,84])<100;"0"→V$[80,80]
288: prt "----------------";spc 2
289: "000000"→V$[91,96];0→A
290: for I=0 to 14;if val(V$[6I+1,6I+6])<val(V$[6I+7,6I+12]);-9→A
291: next I
292: if A#-9;gto +4
293: dsp "NEED DECREASING PRESSURE VALUES";gsb "PEL"
294: prt "PRESSURE VALUES SHOULD DECREASE,PLEASE CORRECT"
295: spc 2;gto -71
296: "ARE PRINTED PRESSURES CORRECT ? "→C$[1,32];gsb "Y/N"
297: if A=-2tn^10;gto -73
298: 0→W;5→R[2];gsb "PRESS CHECK"
299: 15→R[2];gsb "PRESS CHECK"
*11550
```

```
300: 25→R[2];gsb "PRESS CHECK"
301: 40→R[2];gsb "PRESS CHECK"
302: 60→R[2];gsb "PRESS CHECK"
303: 100→R[2];gsb "PRESS CHECK"
304: 120→R[2];gsb "PRESS CHECK"
305: if W#-9;gto +67
306: dsp "PRESS CALIBRATION TAPE IS NEEDED";gsb "BBB"
307: prt "A PRESSURE TAPE IS NEEDED FOR    THIS RADIOSONDE"
308: trk 0;rcf 9,P$,Q$,O$,L$,N$,G$,D$
309: trk 1;ldp 2,0,1
310: dim A$[100],E$[16],B$[100],R[15],C$[32]
311: dim W$[56],U$[895],V$[96],B[25]
312: dim P$[102],Q$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
313: trk 0;ldf 9,P$,Q$,O$,L$,N$,G$,D$
314: trk 1;ldf 0,W$,U$,V$,B[*]
315: gto +44
316: dsp "TURN-ON REMEX & LOAD PRESS TAPE";gsb "BBB"
317: "IS TAPE LOADED & IS DOOR CLOSED?"→C$[1,32];gsb "Y/N"
318: if A=-2tn^10;gto -1
319: if not ios2;gto -3
320: dsp "PROCESSING PRESSURE TAPE";gsb "BBB"
321: time 500;wtc 2,1
322: par 0;red 2,A$;len(A$)→A
323: if A<30;gto -1
324: A$[A-29,A]→B$[1,30]
325: if B$[1,5]#"99999" and B$[1,3]#"499";gto -4
326: val(B$[8,10])+val(B$[12,15])+val(B$[17,20])→A;if A>10000;A-10000→A
327: if A=val(B$[27,30]);0→N→L;gto +4
328: dsp "NEW PRESSURE TAPE IS REQUIRED";gsb "BBB"
329: prt "PRESSURE TAPE ISDEFECTIVE-----"
330: wtc 2,0;spc 2;stp
331: "PT SN    "→A$[1,8];B$[8,10]→A$[9,11];"-"→A$[12];B$[12,15]→A$[13,16]
332: prt A$[1,16]
333: for M=0 to 17
334: par 2;red 2,A$;len(A$)→A
335: if A<71;gto -1
336: A$[A-68,A]→B$[1,69];A$[A-5,A]→B$[65,70]
337: for K=0 to 9;val(B$[1+7K,7+7K])10→Q;if Q<100000;str(prnd(Q,0))→A$[1
338: if Q<10000;str(prnd(Q,0))→A$[2,6];"0"→A$[2,2]
339: if Q<1000;str(prnd(Q,0))→A$[3,6];"00"→A$[2,3]
340: if Q<100;str(prnd(Q,0))→A$[4,6];"000"→A$[2,4]
341: if Q<10;str(prnd(Q,0))→A$[5,6];"0000"→A$[2,5]
342: if Q<1;str(prnd(Q,0))→A$[6,6];"00000"→A$[2,6]
343: if K=0 and N=0;gto +4
344: A$[2,6]→U$[1+5N,5+5N]
345: val(U$[1+5N,5+5N])/10+L→L;if L>10000;L-10000→L
346: N+1→N
347: next K
348: next M
349: par 2;red 2,A$;len(A$)→A
*1883
```

```
350: if A<17;gto -1
351: A$[A-16,A]→B$[1,17]
352: val(B$[12,17])→A
353: wtc 2,0;dsp "CONSERVE ENERGY  TURN-OFF REFEX";gsb "BEP"
354: fxd 1;if A=L;gto +3
355: dsp "NEW PRESSURE TAPE IS REQUIRED";gsb "BEP"
356: prt "PRESSURE TAPE ISDEFECTIVE----";spc 2;stp
357: dsp "PRESSURE TAPE IS GOOD";gsb "BEP"
358: prt "UNDAMAGED TAPE"
359: for I=1 to 96;"0"→V$[I];next I
360: 1→I;0→J;if val(U$[5I-4,5I])#0;gsb "V$"
361: for I=5 to 30 by 5;if val(U$[5I-4,5I])#0;gsb "V$"
362: next I
363: for I=40 to 60 by 10;if val(U$[5I-4,5I])#0;gsb "V$"
364: next I
365: for I=80 to 140 by 20;if val(U$[5I-4,5I])#0;gsb "V$"
366: next I
367: for I=178 to 0 by -1;if val(U$[5I-4,5I])=0;next I
368: I→B[14]→B[9];gsb "V$"
369: gto +3
370: "V$":U$[5I-4,5I]→V$[6J+1,6J+4];"."→V$[6J+5,6J+5]
371: U$[5I,5I]→V$[6J+6,6J+6];1+J→J;ret
372: "ENTER SEC PRESS (650.0 to 1100) "→C$[1,32];gsb "DI"
373: L→B[1]→B[15];if L<650 or L>1100;gto -1
374: spc 2;prt "PRESS LOCK-IN","     SEC:",B[15]
375: "ENTER CONTACT BELOW SEC PRESSURE"→C$[1,32];gsb "DI"
376: L→B[18];L+1→B[20];if L<1 or L>99;gto -1
377: fxd 0;prt " Lo CONT:",B[18]
378: "ENTER PRESSURE AT CONTACT"→C$[1,25];fxd 0;str(B[18])→C$[26];gsb "DI"
379: L→B[17];if L<650;gto -4
380: fxd 1;prt "   PRESS:",B[17]
381: "ENTER PRESSURE AT CONTACT"→C$[1,25];fxd 0;str(B[20])→C$[26];gsb "DI"
382: L→B[19];if L<650;gto -1
383: fxd 0;prt " Hi CONT:",B[20]
384: fxd 1;prt "   PRESS:",B[19];spc 2
385: "ARE PRINTED VALUES OK ?          "→C$[1,32];gsb "Y/N"
386: if A=-2to^10;prt "---REPEATING----";gto -14
387: if B[15]<B[17] and B[15]>=B[19];gto +3
388: dsp "PRESSURES AT SEC ARE INCORRECT";gsb "BEP"
389: prt "---REPEATING----";gto -17
390: (B[17]-B[15])/(B[17]-B[19])+B[18]→B[16]→B[2]→B[10];gsb "PRESS COMP"
391: if B[15]<=B[8]+B[15].0025+.5 and B[15]>=B[8]-B[15].0025-.5;gto +2
392: gto 306
393: dsp "PRESSURE CALIBRATION IS OKAY";gsb "BEP"
394: prt "----------------CALIBRATION OF  PRESSURE IS OKAY"
395: prt "----------------"
396: Y→1→B[24];B[15]/B[8]→B[23]
397: spc 1;prnd(B[16],-1)→B[16];fxd 1;prt " SEC CONT:",B[16];spc 2
398: dsp "PERFORM BASELINE CALIBRATION";gsb "BEP"
399: trk 0;ldf 5
*3481
```

```
400: "BEEP":beep;wait 150;beep;wait 150;beep;wait 3000;ret
401: end
*5530
```

(2) Baseline,Launch
TRACK 0 FILE 5

```
0: dim K[23,2],E[14,15],X[2]
1: dim A[11]
2: trk 0;ldf 4,A[*]
3: gto +68
4: "PRESS COMP":if R[2]>=0;0→J;4→I;0→Y
5: if R[2]>5;1→J;9→I;1→Y
6: if R[2]>10;1→J;14→I;2→Y
7: if R[2]>15;1→J;19→I;3→Y
8: if R[2]>20;1→J;24→I;4→Y
9: if R[2]>25;1→J;29→I;5→Y
10: if R[2]>30;1→J;39→I;6→Y
11: if R[2]>40;1→J;49→I;7→Y
12: if R[2]>50;1→J;59→I;8→Y
13: if R[2]>60;1→J;79→I;9→Y
14: if R[2]>80;1→J;99→I;10→Y
15: if R[2]>100;1→J;119→I;11→Y
16: if R[2]>120;1→J;139→I;12→Y
17: if R[2]>140;1→J;E[14]-1→R[9]→I;13→Y
18: val(U$[5J+1,5J+5])→R[4];val(U$[5I+1,5I+5])→R[3];int(R[2])→A
19: frc(R[2])→J;if A=0;1→A
20: (val(U$[5A-4,5A])-val(U$[5A+1,5A+5]))(-J)+val(U$[5A-4,5A])→R[5]
21: val(V$[6Y+1,6Y+6])→R[6];val(V$[6Y+7,6Y+12])→R[7]
22: R[6]-(R[6]-R[7])(R[4]-R[5])/(R[4]-R[3])→R[8]
23: ret
24: "HUM COMP":(Z+56)/8→R[1];int(R[1])→R[2];frc(R[1])→R[3]
25: D/4+1→R[4];int(R[4])→R[5];frc(R[4])→R[6]
26: K[R[5],2]+(K[R[5]+1,2]-K[R[5],2])R[6]→R[7]
27: R[7]-R[25]→R[8]
28: for N=2 to 13;if R[8]<E[N,R[2]];next N
29: E[N-1,R[2]]+(E[N-1,R[2]+1]-E[N-1,R[2]])R[3]→R[9]
30: E[N,R[2]]+(E[N,R[2]+1]-E[N,R[2]])R[3]→R[10]
31: (R[8]-R[9])/(R[10]-R[9])→R[11]
32: 8*(N-1)+8*R[11]→R[12]
33: if R[12]>100;100→R[12]
34: if R[12]<5;5→R[12]
35: ret
36: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
37: "DI":fxd 1;2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep;dsp C$[1,32]
38: ent "",A$;if A$[1,1]="0" or A$[1,1]="1" or A$[1,1]="2";gto +7
39: if A$[1,1]="3" or A$[1,1]="4" or A$[1,1]="5" or A$[1,1]="6";gto +6
40: if A$[1,1]="7" or A$[1,1]="8" or A$[1,1]="9" or A$[1,2]=".0";gto +5
41: if A$[1,2]=".1" or A$[1,2]=".2" or A$[1,2]=".3" or A$[1,2]=".4";gto +4
42: if A$[1,2]=".5" or A$[1,2]=".6" or A$[1,2]=".7" or A$[1,2]=".8";gto +3
43: if A$[1,2]=".9" or A$[1,1]="-";gto +2
44: gto -7
45: if A#2;gto -8
46: if A$[3,3]="*" or A$[4,4]="*" or A$[5,5]="*" or A$[6,6]="*";sfg 11
47: val(A$)→L;str(L)→A$;A$[2]→A$[1];if L<-90 or L>1100;gto -10
48: "                                "→C$[1,32]
49: ret
*15557
```

```
50: "Y/N":2→A;beep;dsp C$[1,32];ent "",A
51: if A#-2tn^10 and A#-1tn^10;gto -1
52: "                                    "→C$[1,32]
53: ret
54: "DELETE":dsp "ACTIVATED WITH FIRST SIG LEVEL";gsb "BBB"
55: ret
56: "CORRECTING PD":for N=0 to 10;if B[4]<val(G$[5+N16,6+16N]);next N
57: val(G$[5+16(N-1),6+16(N-1)])→B[6]
58: val(G$[5+16N,6+16N])→B[7]
59: val(G$[12+16(N-1),16+16(N-1)])→B[8]
60: val(G$[12+16N,16+16N])→B[9]
61: ((B[4]-B[7])/(B[6]-B[7]))(B[8]-B[9])+B[8]→B[10]
62: ret
63: "TEMP COMP Wx BUR":exp(16.0082991-.99662561n(2B[2]))-48000→B[1]
64: .00046771n(X[1]*B[1]/14000)+.00001113ln(X[1]*B[1]/14000)^2→A
65: 1/(.0033003+A)→A
66: A-273.16→B[5]
67: ret
68: "TEMP COMP":((106.85/B[2]-1.125)tn^5)^.19→B[6]
69: B[12]/B[6]-170→B[5]
70: ret
71: if P$[102,102]="B";gto +91
72: for N=1 to 23;4N-4→K[N,1];next N
73: -735→K[1,2];0→K[2,2];384→K[3,2];625→K[4,2];800→K[5,2];945→K[6,2]
74: 1070→K[7,2];1184→K[8,2];1289→K[9,2];1384→K[10,2];1479→K[11,2]
75: 1568→K[12,2];1645→K[13,2];1741→K[14,2];1830→K[15,2];1919→K[16,2]
76: 2016→K[17,2];2119→K[18,2];2232→K[19,2];2359→K[20,2];2509→K[21,2]
77: 2692→K[22,2];2952→K[23,2]
78: for N=1 to 13;8N→E[N,1];next N
79: 1→N;2745→E[N,2];2737→E[N,3];2727→E[N,4];2726→E[N,5];2724→E[N,6]
80: 2→N;2675→E[N,2];2667→E[N,3];2665→E[N,4];2658→E[N,5];2652→E[N,6]
81: 3→N;2601→E[N,2];2604→E[N,3];2600→E[N,4];2599→E[N,5];2596→E[N,6]
82: 4→N;2510→E[N,2];2499→E[N,3];2512→E[N,4];2509→E[N,5];2508→E[N,6]
83: 5→N;2360→E[N,2];2374→E[N,3];2382→E[N,4];2390→E[N,5];2395→E[N,6]
84: 6→N;2134→E[N,2];2158→E[N,3];2188→E[N,4];2204→E[N,5];2222→E[N,6]
85: 7→N;1861→E[N,2];1904→E[N,3];1949→E[N,4];1983→E[N,5];2021→E[N,6]
86: 8→N;1505→E[N,2];1562→E[N,3];1631→E[N,4];1700→E[N,5];1757→E[N,6]
87: 9→N;1010→E[N,2];1090→E[N,3];1205→E[N,4];1320→E[N,5];1421→E[N,6]
88: 10→N;422→E[N,2];495→E[N,3];637→E[N,4];824→E[N,5];998→E[N,6]
89: 11→N;91→E[N,2];115→E[N,3];179→E[N,4];307→E[N,5];498→E[N,6]
90: 12→N;-10→E[N,2];10→E[N,3];28→E[N,4];88→E[N,5];190→E[N,6]
91: 13→N;-28→E[N,2];-28→E[N,3];-28→E[N,4];-16→E[N,5];-6→E[N,6]
92: 14→N;-40→E[N,2];-32→E[N,3];-24→E[N,4];-16→E[N,5];-8→E[N,6]
93: 14→N;0→E[N,7];8→E[N,8];16→E[N,9];24→E[N,10];32→E[N,11]
94: 13→N;31→E[N,7];51→E[N,8];81→E[N,9];117→E[N,10];156→E[N,11]
95: 12→N;299→E[N,7];389→E[N,8];461→E[N,9];521→E[N,10];582→E[N,11]
96: 11→N;673→E[N,7];788→E[N,8];875→E[N,9];942→E[N,10];1004→E[N,11]
97: 10→N;1117→E[N,7];1198→E[N,8];1266→E[N,9];1322→E[N,10];1378→E[N,11]
98: 9→N;1499→E[N,7];1558→E[N,8];1608→E[N,9];1650→E[N,10];1689→E[N,11]
99: 8→N;1804→E[N,7];1868→E[N,8];1883→E[N,9];1920→E[N,10];1954→E[N,11]
*27370
```

```
100: 7→N;2052→E[N,7];2079→E[N,8];2107→E[N,9];2130→E[N,10];2154→E[N,11]
101: 6→N;2248→E[N,7];2261→E[N,8];2270→E[N,9];2288→E[N,10];2295→E[N,11]
102: 5→N;2401→E[N,7];2408→E[N,8];2411→E[N,9];2418→E[N,10];2422→E[N,11]
103: 4→N;2510→E[N,7];2506→E[N,8];2506→E[N,9];2506→E[N,10];2506→E[N,11]
104: 3→N;2595→E[N,7];2591→E[N,8];2588→E[N,9];2585→E[N,10];2580→E[N,11]
105: 2→N;2651→E[N,7];2648→E[N,8];2641→E[N,9];2638→E[N,10];2630→E[N,11]
106: 1→N;2719→E[N,7];2712→E[N,8];2708→E[N,9];2704→E[N,10];2692→E[N,11]
107: 2685→E[1,12];2625→E[2,12];2569→E[3,12];2506→E[4,12];2429→E[5,12]
108: 2306→E[6,12];2178→E[7,12];1985→E[8,12];1725→E[9,12];1428→E[10,12]
109: 1060→E[11,12];669→E[12,12];255→E[13,12];40→E[14,12]
110: for N=1 to 13;E[N,12]→E[N,13]→E[N,14]→E[N,15];next N
111: 48→E[14,13];56→E[14,14];64→E[14,15]
112: gto +35
113: "BASVAL":dsp "PERFORM BASELINE CALIBRATION";gsb "BBB"
114: "ENTER BASELINE DRY BULB TEMP    "→C$[1,32];gsb "DI"
115: L→T;if L<-90 or L>60;gto -1
116: "ENTER BASELINE WET BULB TEMP    "→C$[1,32];gsb "DI"
117: L→D;if L<-90 or L>60;gto -1
118: 6.11*10^(7.5T/(237.3+T))→V
119: tn^((8.286D+186.527)/(237.3+D))→P
120: P-.00066B[15]abs(T-D)(1+.00115D)→P
121: 100P/V→H;if H<0;0→H
122: if H>100;100→H
123: "ENTER BASELINE REFERENCE R.D.   "→C$[1,32];gsb "DI"
124: L→R[15];if L>94 and L<96;gto +3
125: dsp "Adjust the REFERENCE ADJUST";gsb "BBB"
126: gto -3
127: "ENTER BASELINE TEMP RECORDER DIV"→C$[1,32];gsb "DI"
128: L→R[4];if L<2 or L>90;gto -1
129: "ENTER BASELINE HUM RECORDER DIV "→C$[1,32];gsb "DI"
130: L→R[5];if L<2 or L>90;gto -1
131: prt "CALIBRATION DATAFOR BASELINE:"
132: prt " DRY TEMP",T," WET TEMP",D,"   REF RD",R[15]
133: prt "  TEMP RD",R[4]
134: prt "   HUM RD",R[5];spc 2
135: "ARE PRINTED BASELINE VALUES OK? "→C$[1,32];gsb "Y/N"
136: if A=-2tn^10;prt "----------------REPEAT BASELINE--------";gto -23
137: (95/R[15])R[4]→B[4];gsb "CORRECTING RD"
138: R[4]+B[10]→R[4]
139: (95/R[15])R[5]→B[4];gsb "CORRECTING RD"
140: R[5]+B[10]→R[7]
141: prt "VALUES CORRECTEDFOR DRIFT AND   LINEARITY ARE:"
142: prt " DRY TEMP",T
143: prt " HUMIDITY",H
144: prt "  TEMP RD",R[4]
145: prt "   HUM RD",R[7]
146: ret
147: gsb "BASVAL"
148: T→B[4];H→B[11];R[4]→B[1]
149: (106.85/B[1]-1.125)tn^5→B[2];(B[4]+170)B[2]^.19→B[12]→X[1]
*709
```

```
150: 66.5→B[2];gsb "TEMP COMP"
151: B[5]→B[3]
152: 68.9→B[2];gsb "TEMP COMP"
153: B[5]→B[4]
154: if B[3]<20 or B[4]>30 or H<20 or H>70;gto +3
155: prt "USE THEM TO     LOCK-IN CP-223";spc 1
156: dsp "BASELINE CALIBRATION IS GOOD";gto +6
157: if B[3]<20 or B[4]>30;prt "NO TEMP LOCK-IN IF ARMY AN/AMT-4";gto +3
158: if H<20;prt "HUM UNDER 20%!"
159: if H>70;prt "HUM ABOVE 70%!"
160: dsp "REPEATING BASELINE CALIBRATION";gsb "BBB"
161: prt "----------------REPEAT BASELINE ---------------";gto -14
162: if P$[102,102]="4";gto +62
163: for N=1 to 23;4N-4→K[N,1];next N
164: -28700→K[1,2];-22700→K[2,2];-17700→K[3,2];-13700→K[4,2]
165: -10962→K[5,2];-9017→K[6,2]
166: -7522→K[7,2];-6180→K[8,2];-4967→K[9,2];-3846→K[10,2];-2770→K[11,2]
167: -1736→K[12,2];-748→K[13,2];245→K[14,2];1264→K[15,2];2298→K[16,2]
168: 3394→K[17,2];4567→K[18,2];5831→K[19,2];7234→K[20,2];8852→K[21,2]
169: 10977→K[22,2];13000→K[23,2]
170: for N=1 to 13;8N→E[N,1];next N
171: 1→N;18659→E[N,2];18577→E[N,3];18490→E[N,4];18395→E[N,5];18293→E[N,6]
172: 2→N;17038→E[N,2];16968→E[N,3];16885→E[N,4];16796→E[N,5];16704→E[N,6]
173: 3→N;15740→E[N,2];15674→E[N,3];15627→E[N,4];15601→E[N,5];15567→E[N,6]
174: 4→N;14707→E[N,2];14722→E[N,3];14731→E[N,4];14737→E[N,5];14732→E[N,6]
175: 5→N;13086→E[N,2];13155→E[N,3];13242→E[N,4];13335→E[N,5];13427→E[N,6]
176: 6→N;10887→E[N,2];11092→E[N,3];11308→E[N,4];11532→E[N,5];11778→E[N,6]
177: 7→N;8007→E[N,2];8337→E[N,3];8689→E[N,4];9034→E[N,5];9352→E[N,6]
178: 8→N;4793→E[N,2];5298→E[N,3];5805→E[N,4];6285→E[N,5];6750→E[N,6]
179: 9→N;717→E[N,2];1291→E[N,3];1919→E[N,4];2560→E[N,5];3177→E[N,6]
180: 10→N;-4150→E[N,2];-3602→E[N,3];-2973→E[N,4];-2279→E[N,5];-1562→E[N,6]
181: 11→N;-8708→E[N,2];-8077→E[N,3];-7407→E[N,4];-6693→E[N,5];-5940→E[N,6]
182: 12→N;-13082→E[N,2];-12620→E[N,3];-12161→E[N,4];-11703→E[N,5]
183: -11234→E[N,6]
184: 13→N;-17480→E[N,2];-17134→E[N,3];-16815→E[N,4];-16499→E[N,5]
185: -16210→E[N,6]
186: 14→N;-40→E[N,2];-32→E[N,3];-24→E[N,4];-16→E[N,5];-8→E[N,6]
187: 14→N;0→E[N,7];8→E[N,8];16→E[N,9];24→E[N,10];32→E[N,11]
188: 13→N;-16077→E[N,7];-16037→E[N,8];-16016→E[N,9];-16000→E[N,10]
189: -16030→E[N,11]
190: 12→N;-10773→E[N,7];-10281→E[N,8];-9770→E[N,9];-9258→E[N,10];-8778→E[N,11]
191: 11→N;-5157→E[N,7];-4387→E[N,8];-3687→E[N,9];-3078→E[N,10];-2547→E[N,11]
192: 10→N;-907→E[N,7];-290→E[N,8];299→E[N,9];837→E[N,10];1267→E[N,11]
193: 9→N;3768→E[N,7];4294→E[N,8];4757→E[N,9];5159→E[N,10];5501→E[N,11]
194: 8→N;7211→E[N,7];7619→E[N,8];7985→E[N,9];8313→E[N,10];8604→E[N,11]
195: 7→N;9689→E[N,7];9999→E[N,8];10292→E[N,9];10555→E[N,10];10767→E[N,11]
196: 6→N;11967→E[N,7];12153→E[N,8];12327→E[N,9];12481→E[N,10];12618→E[N,11]
197: 5→N;13498→E[N,7];13555→E[N,8];13616→E[N,9];13680→E[N,10];13721→E[N,11]
198: 4→N;14701→E[N,7];14694→E[N,8];14682→E[N,9];14662→E[N,10];14648→E[N,11]
199: 3→N;15524→E[N,7];15475→E[N,8];15424→E[N,9];15373→E[N,10];15322→E[N,11]
*3231
```

```
200: 2→N;16628→E[N,7];16558→E[N,8];16487→E[N,9];16413→E[N,10];16346→E[N,11]
201: 3→N;18211→E[N,7];18105→E[N,8];18021→E[N,9];17963→E[N,10];17991→E[N,11]
202: 17819→E[1,12];16339→E[2,12];15272→E[3,12];14634→E[4,12];13762→E[5,12]
203: 12768→E[6,12];10979→E[7,12];8900→E[8,12];5851→E[9,12];1097→E[10,12]
204: -2000→E[11,12];-8298→E[12,12];-16000→E[13,12];40→E[14,12]
205: for N=1 to 13;E[N,12]→E[N,13]→E[N,14]→E[N,15];next N
206: 48→E[14,13];56→E[14,14];64→E[14,15]
207: gsb "PASVAL"
208: 9→E[4];8→E[11];P[4]→E[1]
209: exp(16.0082991-.99662561n(2P[1]))-48000→E[6]
210: .0000589986→A;.00247991→B;5.3018981(.00330033-1/(T+273.15))→C
211: (14000/E[6])exp((-B+√(BE-4AC))/2A)→E[12]→X[1]
212: 69.5→P[2];gsb "TEMP COMP Vx IUP"
213: E[5]→E[3]
214: 73.5→P[2];gsb "TEMP COMP Vx IUP"
215: E[5]→E[4]
216: if E[3]<15 or E[4]>35 or H<20 or H>70;gto +3
217: prt "USE THIN TO     LOCK-IN USVP 230and USVP 500";spc 1
218: dsp "BASELINE CALIBRATION IS GOOD";gto +6
219: if E[3]<20 or E[4]>30;prt "NO TEMP LOCK-IN IF NOS 3005!";gto +3
220: if H<20;prt "HUM UNDER 20%!"
221: if H>70;prt "HUM ABOVE 70%!"
222: dsp "REPEATING BASELINE CALIBRATION";gsb "BSB"
223: prt "----------------REPEAT BASELINE ----------------";gto -16
224: prt "----------------BASELINE IS OKAY----------------"
225: (T+56)/8→E[1];int(E[1])→E[2];frc(E[1])→E[3]
226: if E[7]>88;88→E[7]
227: E[7]/4+1→E[4];int(E[4])→E[5];frc(E[4])→E[6]
228: H/8→E[7];int(E[7])→E[8];frc(E[7])→E[9]
229: K[E[5],2]+(K[E[5]+1,2]-K[E[5],2])E[6]→E[10]
230: D[E[8],E[2]]+(D[E[8],E[2]+1]-D[E[8],E[2]])E[3]→E[11]
231: D[E[8]+1,E[2]]+(D[E[8]+1,E[2]+1]-D[E[8]+1,E[2]])E[3]→E[12]
232: E[11]+(E[12]-E[11])E[9]→E[13]
233: E[10]-E[13]→E[25]→X[2]
234: dsp "BASELINE CALIBRATION COMPLETED";gsb "BSB"
235: dsp "      LAUNCH THE FLIGHT..... ";gsb "TRP"
236: prt "FLIGHT LAUNCHED ----------------";spc 2
237: if val(P$[6,6])=9;45→A;gto +2
238: val(P$[7,9])/10→A
239: 1-.00259cos(2A(1-3.14tn^(-7)(10val(P$[19,21]))))→E[22]
240: prt "SFC WIND VALUES AT TIME ""0"":"
241: "ENTER AZIMUTH OFFSET AT LAUNCH  "→C$;gsb "DI"
242: L→A[1];if L<0 or L>=360;gto -1
243: prt " AZ OFFSET",A[1]
244: "ENTER HORIZONTAL DISTANCE OFFSET"→C$;gsb "DI"
245: L→A[2];if L<.1 or L>1500;gto -1
246: prt " HOR DIS",A[2]
247: "ENTER SFC WIND DIRECTION(deg.)"→C$;gsb "DI"
248: L→A[3];if L<0 or L>360;gto -1
249: prt " WIND DIR",A[3]
*13826
```

```
250: "ENTER SFC WIND SPEED"→C$;gsb "DI"
251: L→A[4];if L<0 or L>99;gto -1
252: prt " WIND SPEED",A[4];spc 2
253: "ARE PRINTED SFC WIND VALUES OK ?"→C$[1,32];gsb "Y/N"
254: if A=-2tn^10;prt "---REPEATING----";gto -13
255: "ENTER SURFACE PRESS AT LAUNCH"→C$;gsb "DI"
256: L→A[5]→C;if L<650 or L>1100;gto -1
257: "ENTER SFC CONTACT AT LAUNCH"→C$;gsb "DI"
258: L→B[2];if L>100;gto -1
259: "IS LAUNCH SFC PRESS OK ?"→C$[1,25];fxd 1;str(A[5])→C$[26];gsb "Y/N"
260: if A=-2tn^10;gto -5
261: prt "SFC MET VALUES:"
262: prt " PRESSURE",A[5]
263: str(prnd(A[5],0))→A$;A$[2,4]→P$[22,24]
264: if A[5]>=1000;A$[3,5]→P$[22,24]
265: "IS SFC CONTACT CORRECT ?"→C$[1,24];str(prnd(B[2],-1))→C$[28];gsb "Y/N"
266: if A=-2tn^10;gto -9
267: prt " CONTACT",B[2]
268: B[2]-2.1→R[2];gsb "PRESS COMP"
269: B[24]→I
270: if R[8]>=val(V$[6I+1,6I+6]) or R[8]<=val(V$[6I+7,6I+12]);gto +2
271: R[8]R[23]→R[8]
272: R[8]→B[3]
273: B[2]+2.1→R[2];gsb "PRESS COMP"
274: B[24]→I
275: if R[8]>=val(V$[6I+1,6I+6]) or R[8]<=val(V$[6I+7,6I+12]);gto +2
276: R[8]R[23]→R[8]
277: R[8]→B[4]
278: if A[5]<B[3] and A[5]>B[4];gto +8
279: prt "SFC PRESSURE AT TIME OF LAUNCH  DIDN'T CORRELATEWITH PRESSURE AT"
280: prt "BASELINE;","   THEREFORE,","THIS FLIGHT IS  TERMINATED BUT"
281: prt "IF EITHER A     CONTACT OR PRESSERROR  IS FOUND"
282: prt "THE FLIGHT MIGHTBE SALVAGED BY  RE-ENTERING ALL DATA FOR THIS"
283: prt "FLIGHT"
284: dsp "PREPARE FOR NEXT FLIGHT";gsb "BBB"
285: trk 0;ldp 2
286: str(C/B[23])→A$[1,7];A$[2,7]→W$[1,6]
287: for I=1 to 12
288: if val(W$[1,6])<val(V$[6I+1,6I+6]);next I
289: val(W$[1,6])→R[8];I-1→I;val(V$[6I+1,6I+6])→R[6]
290: val(V$[6I+7,6I+12])→R[7]
291: if I=0;4→I;0→J;gto +11
292: if I=1;9→I;4→J;gto +10
293: if I=2;14→I;9→J;gto +9
294: if I=3;19→I;14→J
295: if I=4;24→I;19→J
296: if I=5;29→I;24→J
297: if I=6;39→I;29→J
298: if I=7;49→I;39→J
299: if I=8;59→I;49→J
*21711
```

```
300: if I=9;79→I;59→J
301: if I=10;99→I;79→J
302: val(U$[5J+1,5J+5])→P[4];val(U$[5I+1,5I+5])→P[3]
303: P[4]-(P[6]-P[8])(P[4]-P[3])/(P[6]-P[7])→P[5]
304: for K=0 to 98;if P[5]<val(U$[5K+1,5K+5]);next K
305: (val(U$[5K-4,5K])-P[5])/(val(U$[5K-4,5K])-val(U$[5K+1,5K+5]))+K→K
306: K-P[2]→P[16]
307: fxd 1;"ENTER SEC REFERENCE RD(tenths)"→C$;gsb "DI"
308: L→P[1]→E;if L<91 or L>99;gto -1
309: if L>94 and L<96;gto +2
310: dsp "adjust the REFERENCE ADJUST";gsb "LPP"
311: prt " REF RD:",P[1]
312: "ENTER SEC TEMPERATURE RD(tenths)"→C$;gsb "DI"
313: L→E[2];if L<1 or L>90;gto -1
314: prt " TEMP RD:",P[2]
315: if P[2]<-40;prt "NO HUMIDITY LOCKIN NEEDED BELOW -40 DEGREES TEMP" .
316: if P[2]<-40;87→P[3];gto +4
317: "ENTER SEC HUMIDITY RD(tenths)"→C$;gsb "DI"
318: L→E[3];if L<1 or L>90;gto -1
319: prt " HUM RD:",B[3];spc 2
320: "ARE RD'S FOR REF,TEMP & HUM OK ?"→C$[1,32];gsb "Y/N"
321: if A=-2tn^10;prt "---REPEATING----";gto -14
322: (95/E[1])E[2]→B[4];gsb "CORRECTING RD"
323: P[4]+P[10]→B[2]
324: if P$[102,102]="4";gsb "TEMP COMP"
325: if P$[102,102]="8";gsb "TEMP COMP WX BUR"
326: B[5]→A[6]
327: (95/E)E[3]→B[4];gsb "CORRECTING RD"
328: E[10]+B[4]→E[3];if E[3]>88;88→E[3]
329: A[6]→Z;B[3]→B;gsb "HUM COMP"
330: P[12]→A[8]
331: 6.11*10^(7.5A[6]/(237.3+A[6]))*(A[8]/100)→E[1]
332: (A[6]+273.16)(1+.376932*(E[1]/P[5]))→A[9]
333: spc 1;prt "AUTO PRESS CONT";fxd 1;prt "CORRECTION",P[16];spc 2
334: 1→A[10];trk 0;rcf 4,A[*]
335: trk 0;rcf 7,W$,US,V$,P[*]
336: trk 0;rcf 8,Y[*],E[*],X[*]
337: "NEED USAF ENG MESSAGE ?"→C$;gsb "Y/N"
338: if A=-2tn^10;"N"→P$[100,100];gto +8
339: if A=-1tr^10;"Y"→P$[100,100]
340: beep;wait 150;beep;ent "ENTER USAF(11111)LOCATION CODE",A$
341: if len(A$)#5;gto -1
342: " USAF CODE:"→C$[1,11];cap(A$[1,5])→C$[12,16];prt C$[1,16];spc 2
343: " IS PRINTED USAF CODE CORRECT ?"→C$[1,32];gsb "Y/N"
344: if A=-2tn^10;gto -7
345: cap(A$[1,5])→C$[1,5]
346: trk 0;rcf 9,P$,Q$,C$,L$,N$,G$,D$
347: trk 0;ldp 6
348: end
*24625
```

c. Flight

TRACK 0 FILE 6

```
0: dim A$[20],E$[36],B$[4],F[12],C$[32]
1: dim A[11]
2: dim W$[56],U$[895],V$[96],B[25]
3: dim K[23,2],E[14,15],X[2]
4: dim F[-1:60,1:6],Y[2]
5: dim H[-2:45,1:6]
6: dim P$[102],O$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
7: trk 0;ldf 4,A[*]
8: trk 0;ldf 7,W$,U$,V$,B[*]
9: trk 0;ldf 8,K[*],E[*],X[*]
10: X[1]→B[12];X[2]→B[25]
11: trk 0;ldf 9,P$,O$,O$,L$,N$,G$,D$
12: if P$[100,100]="N";cfg 12
13: if P$[100,100]="Y";sfg 12
14: sfg 9;if A[10]=1;gto +4
15: trk 0;ldf 10,F[*],Y[*]
16: trk 0;ldf 11,H[*]
17: gto +14
18: 0→A[10];A[11]+1→A[11];trk 0;rcf 4,A[*]
19: .001→H[15,3]
20: B[22]→H[0,3];A[1]→F[-1,1]→H[-1,5];A[2]→F[-1,2]→H[-1,6]
21: A[3]→H[-2,5];A[4]→H[-2,6];A[5]→F[0,4]→H[-1,2];log(A[5])→H[0,2]
22: B[24]→I;A[6]→F[0,1];A[8]→F[0,2];A[9]→F[0,3]→H[-1,4];1→H[1,3]
23: X[1]→B[12];X[2]→B[25];A[1]→H[7,3];.1→H[4,3]
24: 0→H[0,1];200→H[1,1];400→H[2,1];500→H[3,1];600→H[4,1];800→H[5,1]
25: 6→I;for J=1000 to 5000 by 500;J→H[I,1];I+1→I;next J;15→I
26: for J=6000 to 20000 by 1000;J→H[I,1];I+1→I;next J;30→I
27: for J=22000 to 50000 by 2000;J→H[I,1];I+1→I;next J
28: 55000→H[45,1]
29: trk 0;rcf 10,F[*],Y[*]
30: trk 0;rcf 11,H[*]
31: H[1,3]→P;Y[2]→I;for T=Y[1] to 60;T→Y[1]→G
32: gsb "TRUNK"
33: next T
34: dsp "ONLY 60 SIG LEVELS ALLOWED";gsb "BBB"
35: prt "SIG LEVELS USED";gsb "ACTIVE"
36: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
37: "ACTIVE":
38: dsp """DELETE"","OUTPUT","INFO"" ACTIVE";gsb "BBB"
39: sto
40: gto -2
41: ret
42: "DELETE":for I=G-1 to 1 by -1;0→F[G,3];if F[I,3]=0;next I
43: I→Y[1];if I=0;1→Y[1]→Y[2]
44: for I=Y[1]-1 to 0 by -1;I→Y[2];if F[I,3]#0;gto +2
45: next I
46: for N=1 to 45;if H[N,4]>F[Y[1]-1,5];0→H[N,4]→H[N,2];gto +2
47: next N
48: for N=1 to 45;if H[N,4]#0;next N
49: N→P→H[1,3];if Y[1]=1;1→Y[2]
*4553
```

```
50: dsp "DELETING SIGNIFICANT LEVEL";gsb "BBP"
51: prt "----------------REPEATING LEVEL ----------------"
52: if H[5,3]>F[Y[1]-1,5];0→H[5,3]→H[6,3]→H[7,3]→H[11,3]→H[14,3];.001→H[8,3]
53: if H[1,3]=1;.001→H[8,3]→H[15,3];H[-1,5]→H[7,3]
54: .1→H[4,3];0→H[3,3]
55: gto 31
56: ret
57: "DI":fxd 1;2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep;dsp C$[1,32]
58: ent "",A$
59: if A$[1,7]="-2tn^10";gsb "ACTIVE"
60: if A$[1,1]="0" or A$[1,1]="1" or A$[1,1]="2";gto +7
61: if A$[1,1]="3" or A$[1,1]="4" or A$[1,1]="5" or A$[1,1]="6";gto +6
62: if A$[1,1]="7" or A$[1,1]="8" or A$[1,1]="9" or A$[1,2]=".0";gto +5
63: if A$[1,2]=".1" or A$[1,2]=".2" or A$[1,2]=".3" or A$[1,2]=".4";gto +4
64: if A$[1,2]=".5" or A$[1,2]=".6" or A$[1,2]=".7" or A$[1,2]=".8";gto +3
65: if A$[1,2]=".9";gto +2
66: gto -9
67: if A#2;gto -10
68: if A$[3,3]="*" or A$[4,4]="*" or A$[5,5]="*" or A$[6,6]="*";sfg 11
69: val(A$)→L;str(L)→A$;A$[2]→A$[1]
70: "                                    "→C$[1,32]
71: ret
72: "Y/N":2→A;beep;dsp C$[1,32];ent "",A
73: if A#-2tn^10 and A#-1tn^10;gto -1
74: "                                    "→C$[1,32]
75: ret
76: "AWS CONTACT#":
77: if F[T-1,4]>100;"10000850070005000400030002500200015001 00"→W$[1,40]
78: if F[T-1,4]<=100;"0070005000300020001000070005000300020001"→W$[1,40]
79: for W=0 to 9
80: if F[T-1,4]>val(W$[W4+1,W4+4]);gto +2
81: next W
82: for I=1 to 14
83: if val(W$[W4+1,W4+4])<val(V$[6I+1,6I+6]);next I
84: if val(V$[6I+1,6I+6])=0 and B[14]<140;B[14]+1→K;ret
85: if I>14;14→I
86: val(W$[W4+1,W4+4])→R[8];I-1→I;val(V$[6I+1,6I+6])→R[6]
87: val(V$[6I+7,6I+12])→R[7]
88: if I=0;4→I;0→J;gto +14
89: if I=1;9→I;4→J;gto +13
90: if I=2;14→I;9→J;gto +12
91: if I=3;19→I;14→J
92: if I=4;24→I;19→J
93: if I=5;29→I;24→J
94: if I=6;39→I;29→J
95: if I=7;49→I;39→J
96: if I=8;59→I;49→J
97: if I=9;79→I;59→J
98: if I=10;99→I;79→J
99: if I=11;119→I;99→J
*20265
```

```
100: if I=12;139→I;119→J
101: if I=13;R[14]-1→R[9]→I;139→J
102: if val(U$[5I+1,5I+5])=0;R[14]-1→I
103: val(U$[5J+1,5J+5])→R[4];val(U$[5I+1,5I+5])→R[3]
104: R[4]-(R[6]-R[8])(R[4]-R[3])/(R[6]-R[7])→R[5]
105: for K=0 to 178;if R[5]<val(U$[5K+1,5K+5]);next K
106: (val(U$[5K-4,5K])-R[5])/(val(U$[5K-4,5K])-val(U$[5K+1,5K+5]))+K→K
107: ret
108: "PRESS COMP":if R[2]>=1;0→J;4→I;0→Y
109: if R[2]>5;I→J;9→I;1→Y
110: if R[2]>10;I→J;14→I;2→Y
111: if R[2]>15;I→J;19→I;3→Y
112: if R[2]>20;I→J;24→I;4→Y
113: if R[2]>25;I→J;29→I;5→Y
114: if R[2]>30;I→J;39→I;6→Y
115: if R[2]>40;I→J;49→I;7→Y
116: if R[2]>50;I→J;59→I;8→Y
117: if R[2]>60;I→J;79→I;9→Y
118: if R[2]>80;I→J;99→I;10→Y
119: if R[2]>100;I→J;119→I;11→Y
120: if R[2]>120;I→J;139→I;12→Y
121: if R[2]>140;I→J;R[14]-1→R[9]→I;13→Y
122: if val(U$[5I+1,5I+5])=0;R[14]-1→I
123: val(U$[5J+1,5J+5])→R[4];val(U$[5I+1,5I+5])→R[3];int(R[2])→A
124: frc(R[2])→J
125: (val(U$[5A-4,5A])-val(U$[5A+1,5A+5]))(-J)+val(U$[5A-4,5A])→R[5]
126: val(V$[6Y+1,6Y+6])→R[6];val(V$[6Y+7,6Y+12])→R[7]
127: R[6]-(R[6]-R[7])(R[4]-R[5])/(R[4]-R[3])→R[8]
128: ret
129: "CORRECTING RD":for N=0 to 10;if B[4]<val(C$[5+16N,6+16N]);next N
130: val(C$[5+16(N-1),6+16(N-1)])→R[6]
131: val(C$[5+16N,6+16N])→R[7]
132: val(C$[12+16(N-1),16+16(N-1)])→R[8]
133: val(C$[12+16N,16+16N])→R[9]
134: ((B[4]-B[7])/(B[6]-B[7]))(B[8]-B[9])+B[8]→R[10]
135: ret
136: "TEMP COMP":((106.85/R[2]-1.125)tn^5)^.19→R[6]
137: B[12]/R[6]-170→R[5]
138: ret
139: "TEMP COMP Wx BUR":exp(16.0082991-.9966256ln(2R[2]))-48000→R[1]
140: .00046771ln(X[1]*R[1]/14000)+.00001113ln(X[1]*B[1]/14000)^2→A
141: 1/(.0033003+A)→A
142: A-273.16→R[5]
143: ret
144: "HUM COMP":(R[T,1]+56)/8→R[1];int(R[1])→R[2];frc(R[1])→R[3]
145: D/4+1→R[4];int(R[4])→R[5];frc(R[4])→R[6]
146: K[R[5],2]+(K[R[5]+1,2]-K[R[5],2])R[6]→R[7]
147: R[7]-D[25]→R[8]
148: for N=2 to 13;if R[8]<E[N,R[2]];next N
149: E[N-1,R[2]]+(E[N-1,R[2]+1]-E[N-1,R[2]])R[3]→R[9]
*1337
```

```
150: E[N,R[2]]+(E[N,R[2]+1]-E[N,R[2]])R[3]→R[10]
151: (R[8]-R[9])/(R[10]-R[9])→R[11]
152: 8*(N-1)+8*R[11]→R[12]
153: if R[12]>100;100→R[12]
154: if R[12]<5;5→R[12]
155: ret
156: "TRUNK":
157: if flg9;cfg 9;gto +76
158: fxd 0;str(T)→A$;A$[2]→A$[1]
159: dsp "EVALUATE SIGNIFICANT LEVEL #",A$[1,3];gsb "BBB"
160: spc 1;prt "SIG LEVEL#",val(A$[1,3])
161: if F[T-1,4]<=1;cfg 12
162: if not flg12;"ENTER PRESS CONTACT (tenths)"→C$;gto +5
163: gsb "AWS CONTACT#"
164: fxd 1;str(rnd(K-B[16],-1))→A$;A$[2]→A$[1]
165: "ENTER PRESS CONTACT:("→C$[1,21];A$[1,5]→C$[22,26];len(A$)→C
166: ")"→C$[22+C,22+C];"     "→C$[23+C,27+C]
167: gsb "DI"
168: L→B;if L+B[16]<1 or L+B[16]>180;gto -6
169: if L+B[16]>B[14];spc 2;prt "FLIGHT FINISHED";spc 2;gsb "ACTIVE"
170: if flg12=0;gto +5
171: if rnd(L,-1)<=rnd(K-B[16],-1);gto +3
172: dsp "ENTER MANDATORY CONTACT";gsb "BBB"
173: gto -10
174: if rnd(K-B[16],-1)=rnd(L,-1);val(W$[W4+1,W4+4])→F[T,4];gto +6
175: L+B[16]→R[2];gsb "PRESS COMP"
176: B[24]→I
177: if R[8]>=val(V$[6I+1,6I+6]) or R[8]<=val(V$[6I+7,6I+12]);gto +2
178: R[8]B[23]→R[8]
179: R[8]→F[T,4]
180: if F[T-1,4]>F[T,4];gto +10
181: fxd 1;dsp "PREVIOUS CONTACT GTR THAN",L;gsb "BBB"
182: gto -20
183: "IS MET DATA MISSING AT CONTACT ?"→C$[1,32];gsb "Y/N"
184: if A=-2tn^10;gto +6
185: prt " TEMP & HUMIDITY ARE MISSING AT  THIS MANDATORY"
186: fxd 0;prt " PRESS LEVEL",F[T,4]
187: for N=3 to 6;if F[-1,N]=0;F[T,4]→F[-1,N];ret
188: next N
189: ret
190: fxd 1;prt " CONTACT",R
191: "ENTER TIME (0.01 to 150 minutes)"→C$;gsb "DI"
192: L→F[T,5];if L<.01 or L>150;gto -1
193: if L>F[T-1,5];gto +3
194: fxd 2;dsp "PREVIOUS SIG LEV TIME WAS",F[T-1,5];gsb "BBB"
195: gto -4
196: fxd 2;prt " TIME",F[T,5]
197: cfg 11;"ENTER REFERENCE ORDINATE(tenths)"→C$;gsb "DI"
198: L→B[1]→Z;if L<91 or L>99;gto -1
199: if L>94 and L<96;gto +2
*6749
```

```
200: dsp "adjust the REFERENCE ADJUST";gsb "BBB"
201: if not flg11;gto +4
202: "DOES MISSING DATA STOP HERE ?"→C$;gsb "Y/N"
203: if A=-2tn^10;gto -20
204: if A=-1tn^10;spc 1;prt "MET DATA RESUMESHAD BEEN MISSING";spc 1
205: prt " REF RD",B[1]
206: "ENTER TEMPERATURE RD (tenths)    "→C$;gsb "DI"
207: L→B[2];if L<1 or L>90;gto -1
208: prt " TEMP RD",B[2]
209: (95/B[1])B[2]→B[4];gsb "CORRECTING RD"
210: B[4]+B[10]→B[2]
211: if P$[102,102]="4";gsb "TEMP COMP"
212: if P$[102,102]="B";gsb "TEMP COMP Wx BUR"
213: B[5]→F[T,1];if B[5]<-40 or F[T,4]<100;gto +8
214: "ENTER HUMIDITY RD (tenths)       "→C$;gsb "DI"
215: L→B[3];if L<1 or L>90;gto -1
216: prt " HUM RD",B[3]
217: (95/Z)B[3]→B[4];gsb "CORRECTING RD"
218: B[4]+B[10]→B[3]→D
219: gsb "HUM COMP"
220: B[12]→F[T,2]
221: spc 2;"IS PRINTED DATA FOR SIG LEV OK ?"→C$[1,32];gsb "Y/N"
222: if A=-2tn^10;prt "---REPEATING----";gto -64
223: 1+C→C
224: 6.11*10^(7.5F[T,1]/(237.3+F[T,1]))→B[1]
225: B[1]*F[T,2]/100→B[1]
226: (F[T,1]+273.16)(1+.376932*(B[1]/F[T,4]))→F[T,3]
227: for I=T-1 to 0 by -1;I→Y[2];T→Y[1];if F[I,3]=0;next I
228: F[I,6]+(log(F[I,4])-log(F[T,4]))67.442((F[I,3]+F[T,3])/2)→F[T,6]
229: prrd((F[T,3]-F[I,3])/.001(F[T,6]-F[I,6]),-1)→A
230: dsp "TEMP LAPSE RATE(d/km):",A
231: trk 0;rcf 10,F[*],Y[*]
232: gto +5
233: if P<=45;gto +3
234: dsp "ONLY 45 ZONE LEVELS ALLOWED";gsb "BBB"
235: prt "ZONE LEVELS USED";gsb "ACTIVE"
236: fxd 0;dsp "OK FOR MESSAGES TO HEIGHT",H[P-1,1];gsb "BBB"
237: H[0,3]H[P,1]6371299/(6371299+H[P,1])→B[8]
238: F[I,6]→B[6];F[T,6]→B[7];if B[7]<B[8];ret
239: F[I,5]+((B[8]-B[6])/(B[7]-B[6]))(F[T,5]-F[I,5])→H[P,4]
240: (2F[T,3]/(F[T,6]-B[8])-(F[T,3]-F[I,3])/(F[T,6]-F[I,6]))33.721→A
241: log(F[T,4])+1/A→H[P,2]
242: if H[P,4]>H[11,3];0→H[3,3]
243: if H[3,3]=2;gto +46
244: if H[4,3]=.1;int(H[P,4]10)/10→H[8,3];H[8,3]+.1→H[11,3]
245: if H[4,3]=.5;int(H[P,4])→H[8,3];int(H[P,4])+.5→H[11,3]
246: if H[4,3]=.5;if frc(H[P,4])>.5;H[8,3]+.5→H[8,3];H[11,3]+.5→H[11,3]
247: if H[4,3]=1;int(H[P,4])→H[8,3];int(H[P,4])+1→H[11,3]
248: if H[8,3]=H[14,3] and H[11,3]=H[15,3];gto +41
249: if H[P,4]<=H[5,3];H[6,3]→H[9,3];H[7,3]→H[10,3];gto +25
*9341
```

```
250: if H[15,3]=H[8,3];H[12,3]→H[9,3];H[13,3]→H[10,3];gto +23
251: fxd 1;if H[3,3]=2;fxd 2
252: cfg 11;"ENTER ELEV ANGLE AT TIME:"→C$[1,25];str(H[8,3])→C$[26]
253: gsb "DI"
254: L→H[9,3];if L<0 or L>90;gto -2
255: if not flg11;gto +13
256: if H[P,1]<=1000;dsp "INACTIVE UNTIL HEIGHT ABOVE 1 KM";gsb "BBB"
257: if H[P,1]<=1000;gto -5
258: if L=.1 or L=.5 or L=1;L→H[4,3];gto -14
259: if L#0;gto -8
260: 2→H[3,3];prt "----------------ANGLES MISSING  ----------------"
261: "ENTER TIME ANGLES BEGIN MISSING"→C$[1,32];gsb "DI"
262: L→H[8,3];if L<.01 or L>150;gto -1
263: "ENTER TIME ANGLES STOP MISSING"→C$[1,32];gsb "DI"
264: L→H[11,3];if L<.01 or L>150;gto -1
265: if H[5,3]>=H[8,3];gto +9
266: if H[8,3]=0;0→H[9,3];H[-1,5]→H[10,3];gto +8
267: gto -16
268: spc 1;prt "TOP OF ZONE DATA","  TIME:",H[8,3]
269: fxd 2;prt " EL ANGLE",H[9,3]
270: fxd 1;if H[3,3]=2;fxd 2
271: "ENTER AZ ANGLE AT TIME:  "→C$[1,25];str(H[8,3])→C$[26];gsb "DI"
272: L→H[10,3];if L<0 or L>360;gto -2
273: fxd 2;prt " AZ ANGLE",H[10,3]
274: fxd 1;if H[3,3]=2;fxd 2
275: "ENTER ELEV ANGLE AT TIME:"→C$[1,25];str(H[11,3])→C$[26]
276: cfg 11;gsb "DI"
277: L→H[12,3];if L<0 or L>90;gto -3
278: if flg11;gto -22
279: prt "  TIME:",H[11,3]
280: fxd 2;prt " EL ANGLE",H[12,3]
281: fxd 1;if H[3,3]=2;fxd 2
282: "ENTER AZ ANGLE AT TIME:  "→C$[1,25];str(H[11,3])→C$[26];gsb "DI"
283: L→H[13,3];if L<0 or L>360;gto -2
284: fxd 2;prt " AZ ANGLE",H[13,3];spc 2
285: if abs(H[9,3]-H[12,3])>50H[4,3];dsp "DOUBLE CHECK DATA";gsb "BBB"
286: if abs(H[10,3]-H[13,3])>50H[4,3];dsp "DOUBLE CHECK DATA";gsb "BBB"
287: "ARE PRINTED ANGLES FOR TIME OK ?"→C$[1,32];gsb "Y/N"
288: if A=-2tn^10;prt "---REPEATING----";gto -37
289: (H[8,3]-H[P,4])/(H[8,3]-H[11,3])→L
290: if H[10,3]-H[13,3]>180;H[13,3]+360→H[13,3]
291: if H[13,3]-H[10,3]>180;H[10,3]+360→H[10,3]
292: H[10,3]-L(H[10,3]-H[13,3])→H[P,5]
293: if H[P,5]>360;H[P,5]-360→H[P,5]
294: H[9,3]-L(H[9,3]-H[12,3])→H[P,6]
295: P+1→P→H[1,3]
296: H[11,3]→H[5,3]→H[15,3];H[12,3]→H[6,3];H[13,3]→H[7,3];H[8,3]→H[14,3]
297: trk 0;rcf 11,H[*]
298: if P#35;gto -65
299: fxd 0;dsp "BALLOON ALT(gr):",F[1,6];gsb "BBB"
*22052
```

```
300: dsp "AUTOMATIC OUTPUT AT ALT OF 30KM";gsb "EPP"
301: spc 2
302: prt "----------------AUTOMATIC OUTPUTAT 30 KILOMETERS----------------"
303: spc 2
304: trk 0;ldp 12
305: end
*5870
```

d. Output.

(1) Sound Ranging, Computer Met, Fallout, Ballistic 3

TRACK 0 FILE 12

```
0: dsp "TURN-ON REMEX";gsb "BBB"
1: dim A$[100],L$[5],F[10],C[0:16,1:14]
2: dim P$[102],Q$[16],O$[16],I$[16],H$[6],C$[192],D$[16]
3: dim F[-1:60,1:6],Y[2]
4: dim H[-2:45,1:6],L[15]
5: dim H$[2,32];time 500
6: trk 0;ldf 9,P$,Q$,O$,L$,N$,C$,D$
7: spc 5;prt "FLIGHT DATE:",D$[1,16]
8: sfg 4;if not ios2;cfg 4
9: spc 5
10: dsp "MET MESSAGE OUTPUT";gsb "BBB"
11: if P$[101,101]="Y";sfg 7;cfg 8
12: if P$[101,101]="N";cfg 7;sfg 8
13: trk 0;ldf 10,F[*],Y[*]
14: trk 0;ldf 11,H[*]
15: " 5 9 #,. )4&80:;3 $? 6!/-2´ 71( "→H$[1]
16: " T O HNM LRGIPCVEZDBSYFXAWJ UQK "→H$[2]
17: char(0)→H$[1,1,1];char(10)→H$[1,9,9];char(7)→H$[1,21,21]
18: char(13)→H$[1,3,3]
19: gsb "INTHT"
20: sfg 3;dsp "SOUND RANGING";gsb "BBB"
21: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
22: gto +69
23: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
24: "INTHT":for N=1 to Y[1];if F[N,3]#0;gto +7
25: for K=N to Y[1];if F[N,3]=0;next K
26: (F[K,5]-F[N,5])/(F[K,5]-F[N-1,5])→C
27: F[K,1]-C(F[K,1]-F[N-1,1])→F[N,1]
28: F[K,2]-C(F[K,2]-F[N-1,2])→F[N,2]
29: F[K,3]-C(F[K,3]-F[N-1,3])→F[N,3]
30: F[K,6]-C(F[K,6]-F[N-1,6])→F[N,6]
31: next N
32: ret
33: "FOOT":fmt 2,"9",/
34: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2;wtc 2,0
35: fmt b,z
36: if flg4 and flg7;par 0;wtc 2,2;wrt 2,3,2,8;wtc 2,0
37: if flg4;wtc 2,2;for N=1 to 30;wtb 2,0;next N;wtc 2,0
38: ret
39: "A→B":fmt b,z;par 0
40: for C=1 to B
41: for K=1 to 2;pos(H$[K],A$[C,C])→F;if F;sfg K;gto +2
42: next K
43: if F=1 or F=3 or F=5 or F=9;cfg 1,2,5,6;gto +5
44: if flg1 and flg5;cfg 1;gto +4
45: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
46: if flg2 and flg6;cfg 2;gto +2
47: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
48: wtc 2,2;wrt 2,F-1;wtc 2,0
49: next C
```

*23236

```
50: wtc 2,2;wrt 2,2,8;wtc 2,0
51: ret
52: "R→P":√(XX+YY)→R[1]
53: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
54: A→R[2]
55: if A<0;360+A→R[2]
56: ret
57: "HEADER":cfg 1,2,5,6;par 2
58: if flg4 and flg8;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
59: if flg4 and flg7;12→B;gsb "A→B"
60: dsp A$[1,12];prt A$[1,12]
61: B$[85,99]→A$[1,J]
62: if flg9;cfg 9;gto +5
63: fmt 3,c12
64: if flg4 and flg8;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
65: if flg4 and flg7;12→B;gsb "A→B"
66: gto +3
67: if flg4 and flg8;fmt 3,c15;wtc 2,2;wrt 2.3,A$[1,15];wtc 2,0
68: if flg4 and flg7;15→B;gsb "A→B"
69: dsp A$[1,J];prt A$[1,J]
70: ret
71: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000
72: ret
73: "P→R":R[1]*cos(R[2])→X;R[1]*sin(R[2])→Y
74: ret
75: "STRING":fxd 0;str(M)→B$[1,5]
76: if M<1000;B$[2,4]→B$[3,5];"0"→B$[2,2]
77: if M<100;B$[2,4]→B$[3,5];"0"→B$[2,2]
78: if M<10;B$[2,4]→B$[3,5];"0"→B$[2,2]
79: if M<1;B$[2,4]→B$[3,5];"0"→B$[2,2]
80: ret
81: "ARRAY":I→M;gsb "STRING"
82: B$[4,5]→A$[1,2];A→M;gsb "STRING"
83: B$[3,5]→A$[3,5];S→M;gsb "STRING"
84: B$[3,5]→A$[6,8];T→M;gsb "STRING"
85: B$[2,5]→A$[9,12];O→M;gsb "STRING"
86: B$[2,5]→A$[13,16]
87: if flg4 and flg8;par 2;fmt 4,c16;wtc 2,2;wrt 2.4,A$[1,16];wtc 2,0
88: if flg4 and flg7;16→B;gsb "A→B"
89: dsp A$[1,16];prt A$[1,16]
90: ret
91: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→R"
92: X→R[4];Y→R[6]
93: prnd(H[-2,5]*16/9,0)→A→R[10]
94: prnd(H[-2,6],0)→S→R[9]
95: prnd(H[-1,4]10,0)→T
96: .0000001→H[0,4]
97: 0→I
98: if H[5,4]=0;spc 5;prt "SOUND RANGING   NOT READY";gto +48
99: for P=1 to 5;if H[P,4]=0;gto +13
*19405
```

```
100: P-1→J
101: if P=3;next P
102: if P=4;2→J
103: I+1→I
104: 6367650→S
105: √((S+R[P,1])^2-S^2cos(R[P,6])^2)-Scin(R[P,6])→R[P]
106: (Scos(R[P,6])/(S+R[P,1]))R[9]→R[1]
107: R[P,5]→R[2];gsb "P→F"
108: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];X+R[6];R[7]-R[6]→Y;gsb "F→P"
109: prnd((R[1]/(R[P,4]-R[J,4])).03230,0)→C→R[1]
110: prnd(R[2]*16/9,0)→A→R[I+4]
111: next P
112: if R[2]<R[1] and (R[2]>=R[9]+2 or R[2]<=R[9]-2);R[2]→C;R[6]→A;gto +16
113: if R[2]<R[1] and (R[2]<R[9]+2 and R[2]>R[9]-2);R[1]→C;R[5]→A;gto +15
114: if R[2]>2*R[1];.4→r1;0→r2;.3→r3;.15→r4;.15→r5;gto +2
115: .2→r1;.5→r2;.15→r3;.075→r4;.075→r5
116: 0→X→Y→C→D
117: r1B[9]→R[1];R[10]*9/16→R[2];gsb "P→F"
118: X→C;Y→D
119: r2B[1]→R[1];R[5]*9/16→R[2];gsb "P→F"
120: X+C→C;Y+D→D
121: r3B[2]→R[1];R[6]*9/16→R[2];gsb "P→F"
122: X+C→C;Y+D→D
123: r4B[3]→R[1];R[7]*9/16→R[2];gsb "P→F"
124: X+C→C;Y+D→D
125: r5B[4]→R[1];R[8]*9/16→R[2];gsb "P→F"
126: X+C→X;Y+D→Y;gsb "F→P"
127: R[1]→S;R[2]*16/9→A
128: prnd(S,0)→S;prnd(A,0)→A
129: if A=0;64→A
130: if S=0;0→A
131: for L=1 to 5;if not (R[L,6]>200 and R[L-1,6]<200);next L
132: (200-R[L-1,6])/(R[L,6]-R[L-1,6])→O
133: O(R[L,3]-R[L-1,3])+R[L-1,3]→R[16,3]
134: O(R[L,1]-R[L-1,1])+R[L-1,1]+273.16+3R[16,3]→R[16,3]
135: R[16,3]/4-273.16→R[16,3]→T
136: prnd(T*10,0)→T
137: abs(T)→Y;" "→A$[1,1];if T<0;"-"→A$[1,1]
138: gsb "SPLIT"
139: B$[3,5]→A$[2,4];A→Y;gsb "SPLIT"
140: B$[3,5]→A$[5,7];S→Y;gsb "SPLIT"
141: B$[4,5]→A$[8,9]
142: if not flg3;gto +5
143: A$[1,9]→B$[91,99];B$[73,84]→A$[1,12];15→J
144: sfg 9;gsb "READER"
145: cfg 9;gsb "PCOT"
146: stc 5
147: sfg 3;gsb "COMPUTER OPS";gsb "OPF"
148: R[-1,5]→R[2];R[-1,6]→R[1];gsb "P→F"
149: X→R[4];Y→R[6]
*1555
```

```
150: str(prnd(H[-1,2],0))→A$;A$[2,4]→P$[22,24]
151: if H[-1,2]>=1000;A$[3,5]→P$[22,24]
152: P$[1,12]→A$[1,12];P$[13,24]→P$[85,96];12→J
153: if not flg3;gto +34
154: gsb "HEADER"
155: 0→I
156: prnd(H[-2,5]*16/9,0)→A;if A=0;64→A
157: prnd(H[-2,6],0)→S;if S=0;0→A
158: prnd(H[-1,4]10,0)→T
159: prnd(H[-1,2],0)→Q
160: gsb "ARRAY"
161: .0000001→H[0,4]
162: for P=1 to 29;if H[P,4]=0;gto +23
163: if P=1;0→J
164: if P=2;next P
165: if P=3;1→J
166: if P=4 or P=5;next P
167: if P=6;P-3→J
168: if P>6;P-1→J
169: 1+I→I
170: 6367650→S
171: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
172: (Scos(H[P,6])/(S+H[P,1]))R[8]→R[1]
173: H[P,5]→R[2];gsb "P→R"
174: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "R→P"
175: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S
176: H[0,3]H[P,1]6371299/(6371299+H[P,1])→B[1]
177: H[0,3]H[J,1]6371299/(6371299+H[J,1])→B[2]
178: prnd(((B[1]-B[2])/67.442(H[J,2]-H[P,2]))10,0)→T
179: tn^((H[P,2]+H[J,2])/2)→Q;prnd(Q,0)→Q
180: prnd(R[2]*16/9,0)→A
181: if A=0;64→A
182: if S=0;0→A
183: gsb "ARRAY"
184: next P
185: gsb "FOOT"
186: spc 5
187: sfg 3;dsp "FALLOUT MET";gsb "BBB"
188: str(prnd(1000H[-1,2]/1013.25,0))→A$;A$[2,4]→P$[22,24]
189: if H[-1,2]>=1013.25;A$[3,5]→P$[22,24]
190: gto +11
191: "ARRAY-12":I→M;gsb "STRING"
192: P$[4,5]→A$[1,2];A→M;gsb "STRING"
193: P$[4,5]→A$[3,4];S→M;gsb "STRING"
194: P$[4,5]→A$[5,6];T→M;gsb "STRING"
195: P$[3,5]→A$[7,9];Q→M;gsb "STRING"
196: P$[3,5]→A$[10,12]
197: if flg4 and flg8;par 2;fmt 3,cl2;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
198: if flg4 and flg7;12→E;gsb "A→E"
199: dsp A$[1,12];prt A$[1,12]
*14516
```

```
200: ret
201: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→R"
202: X→R[4];Y→R[6]
203: 1→I
204: H[-1,2]→C[0,4]
205: for P=1 to 27;if H[P,4]=0;gto +23
206: P-2→J
207: if P=1;0→J
208: if P=2;next P
209: if P=3;1→J
210: if P=4 or P=5;next P
211: if P=6;P-3→J
212: if P=7 or P=8 or P=15;P-1→J
213: if P=9 or P=11 or P=13 or P=16 or P=18;next P
214: if P=20 or P=22 or P=24 or P=26;next P
215: 6371299→S
216: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
217: (Scos(H[P,6])/(S+H[P,1]))R[8]→R[1]
218: H[P,5]→R[2];gsb "P→R"
219: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "P→P"
220: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S→C[I,2]
221: H[0,3]H[P,1]6371299/(6371299+H[P,1])→B[1]
222: H[0,3]H[J,1]6371299/(6371299+H[J,1])→B[2]
223: (B[1]-B[2])/67.442(H[J,2]-H[P,2])→C[I,3];prnd(C[I,3],0)→T
224: tn^((H[P,2]+H[J,2])/2)→Q→C[I,4];prnd(Q,0)→Q
225: R[2]→C[I,5]
226: if I<=14;I+1→I→C[I,1]
227: next P
228: for Z=1 to 16;if C[Z,4]≠0;next Z
229: Z-1→Z
230: H[-1,4]→C[0,3]
231: for N=0 to Z;348.38395C[N,4]/C[N,3]→C[N,9];next N
232: C[0,9]/1225→C[0,10];C[1,9]/1213.3→C[1,10];C[2,9]/1184.4→C[2,10]
233: C[3,9]/1139.2→C[3,10];C[4,9]/1084.6→C[4,10];C[5,9]/1032→C[5,10]
234: C[6,9]/957→C[6,10];C[7,9]/863.4→C[7,10];C[8,9]/777→C[8,10]
235: C[9,9]/697.4→C[9,10];C[10,9]/590→C[10,10];C[11,9]/467→C[11,10]
236: C[12,9]/364.8→C[12,10];C[13,9]/266.6→C[13,10];C[14,9]/194.8→C[14,10]
237: C[15,9]/142.3→C[15,10]
238: 100C[0,10]→C[0,11]
239: H[-1,4]→C[0,3]
240: C[0,3]/288.2→C[0,6]
241: C[1,3]/287.5→C[1,6];C[2,3]/285.9→C[2,6];C[3,3]/283.3→C[3,6]
242: C[4,3]/280→C[4,6];C[5,3]/276.8→C[5,6];C[6,3]/271.9→C[6,6]
243: C[7,3]/265.5→C[7,6];C[8,3]/259→C[8,6];C[9,3]/252.5→C[9,6]
244: C[10,3]/242.7→C[10,6];C[11,3]/229.8→C[11,6];C[12,3]/216.8→C[12,6]
245: C[13,3]/216.7→C[13,6];C[14,3]/216.7→C[14,6];C[15,3]/216.7→C[15,6]
246: gto +9
247: "ARRAY-FO":I→M;gsb "STRINC"
248: B$[4,5]→A$[1,2];A→M;gsb "STRINC"
249: B$[3,5]→A$[3,5];S→M;gsb "STRING"
*7265
```

```
250: P$[3,5]→A$[6,8]
251: if flg4 and flg8;par 2;fmt 1,c8;wtc 2,2;wrt 2.1,A$[1,8];wtc 2,0
252: if flg4 and flg7;8→P;gsb "A→P"
253: dsp A$[1,8];prt A$[1,8]
254: ret
255: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→I"
256: X→R[4];Y→R[6]
257: rnd(H[-2,5]*16/9,0)→A;if A=0;64→A
258: rnd(H[-2,6],0)→S;if S=0;0→A
259: P$[49,60]→A$[1,12];P$[61,72]→P$[85,96];12→J
260: if not flg3;gto +28
261: gsb "HEADER"
262: 0→I
263: gsb "ARRAY-FO"
264: 1→I
265: for P=8 to 34;if H[P,4]=0;gto +21
266: if P=8;0→J
267: if P=9 or P=10 or P=11;next P
268: if P=12;8→J
269: if P=13 or P=14;next P
270: if P=15;12→J
271: if P=16 or P=18 or P=20 or P=22 or P=24 or P=26 or P=28;next P
272: if P=17 or P=19 or P=21 or P=23 or P=25 or P=27 or P=29;P-2→J
273: if P>29;P-1→J
274: 6371299→S
275: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
276: (Scos(H[P,6])/(S+H[P,1]))R[8]→R[1]
277: H[P,5]→R[2];gsb "P→I"
278: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "R→P"
279: rnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S
280: rnd(R[2]*16/9,0)→A
281: if A=0;64→A
282: if S=0;0→A
283: gsb "ARRAY-FO"
284: 1+I→I
285: next P
286: gsb "FOOT"
287: spc 5
288: sfg 3;dsp "BALLISTIC TYPE 3";gsb "BBB"
289: gto +11
290: "T3":
291: E+1→E;for N=1 to Z;rNC[E,6]+C[N,7]→C[N,7];next N
292: ret
293: "D3":
294: E+1→E;for N=1 to Z;rNC[E,10]+C[N,11]→C[N,11];next N
295: ret
296: "WW":
297: C[I,2]→R[1];C[I,5]→R[2];1+I→I;gsb "P→P"
298: for N=1 to Z;XrN+C[N,13]→C[N,13];YrN+C[N,14]→C[N,14];next N
299: ret
*25544
```

```
300: 100C[0,6]→C[0,7]
301: 100→r1;27→r2;13→r3;8→r4;5→r5;4→r6;2→r7;1→r8→r9→r10→r11→r12→r13→r14→r15
302: gsb "T3"
303: 0→r1;73→r2;20→r3;12→r4;10→r5;4→r6→r7;3→r8;gsb "T3"
304: 0→r2;67→r3;25→r4;20→r5;9→r6;7→r7;5→r8;2→r9→r10→r11→r12→r13→r14→r15
305: gsb "T3"
306: 0→r3;55→r4;21→r5;11→r6;9→r7;4→r8;3→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
307: 0→r4;44→r5;13→r6;11→r7;10→r8;gsb "T3"
308: 0→r5;59→r6;26→r7;19→r8;9→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
309: 0→r6;41→r7;23→r8;13→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
310: 0→r7;35→r8;24→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
311: 0→r8;44→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
312: 100→r1;43→r2;22→r3;15→r4;11→r5;8→r6;6→r7;5→r8;4→r9;3→r10;1→r11
313: 2→r12→r13→r14→r15;0→P;gsb "D3"
314: 0→r1;57→r2;31→r3;21→r4;17→r5;11→r6;8→r7;6→r8;6→r9;4→r10;3→r11→r12→r14
315: 4→r15;gsb "D3"
316: 0→r2;47→r3;32→r4;25→r5;17→r6;14→r7;11→r8;9→r9;7→r10;5→r11→r12→r14→r15
317: 4→r13;gsb "D3"
318: 0→r3;32→r4;22→r5;17→r6;13→r7;5→r13;gsb "D3"
319: 0→r4;25→r5;15→r6;12→r7;10→r8;8→r9;6→r11;gsb "D3"
320: 0→r5;32→r6;22→r7;19→r8;17→r9;13→r10;12→r11;11→r12→r13;10→r14→r15;gsb "D3"
321: 0→r6;25→r7;17→r8;15→r9;12→r10;11→r11;10→r12;9→r13→r14→r15;gsb "D3"
322: 0→r7;21→r8;14→r9;11→r10;9→r11→r12;8→r14→r15;gsb "D3"
323: 0→r8;18→r9;11→r10;9→r11;8→r12→r13;7→r14→r15;gsb "D3"
324: 0→r9;25→r10;16→r11;14→r12→r13;13→r14;12→r15;gsb "D3"
325: 0→r10;23→r11;12→r12;10→r13;11→r14;9→r15;gsb "D3"
326: 0→r11;16→r12;9→r13;8→r14→r15;gsb "D3"
327: 0→r12;12→r13;6→r14;5→r15;gsb "D3"
328: 0→r13;8→r14;5→r15;gsb "D3"
329: 0→r14;6→r15;gsb "D3"
330: 1→I
331: 100→r1;20→r2;9→r3;6→r4;4→r5;3→r6;2→r7→r8→r9;1→r10;0→r11→r12→r15;gsb "WW"
332: 0→r1;80→r2;19→r3;12→r4;8→r5;5→r6;3→r7;2→r10;1→r12→r13→r14→r15;gsb "WW"
333: 0→r2;72→r3;26→r4;15→r5;8→r6;7→r7;6→r8;5→r9;1→r11;gsb "WW"
334: 0→r3;56→r4;20→r5;9→r6;4→r10→r11;2→r12;gsb "WW"
335: 0→r4;53→r5;12→r6;8→r7;3→r10→r11;4→r12;3→r13;2→r14→r15;gsb "WW"
336: 0→r5;63→r6;20→r7;14→r8;12→r9;7→r10→r12→r13→r14→r15;8→r11;gsb "WW"
337: 0→r6;53→r7;19→r8;13→r9;8→r10→r11;7→r12;gsb "WW"
338: 0→r7;45→r8;20→r9;9→r10→r11;gsb "WW"
339: 0→r8;36→r9;8→r12;gsb "WW"
340: 0→r9;55→r10;20→r11;17→r12;15→r13;13→r14;12→r15;gsb "WW"
341: 0→r10;38→r11;16→r12;14→r13;gsb "WW"
342: 0→r11;30→r12;13→r13;11→r15;gsb "WW"
343: 0→r12;24→r13;10→r14→r15;gsb "WW"
344: 0→r13;18→r14;8→r15;gsb "WW"
345: 0→r14;14→r15;gsb "WW"
346: for N=1 to Z;C[N,13]→X;C[N,14]→Y;gsb "P→P"
347: P[1]→C[N,13];P[2]→C[N,14];next N
348: H[-2,5]→C[0,14];H[-2,6]100→C[0,13]
349: if not flg3;gto +14
*8627
```

```
350: P$[25,36]→A$[1,12];P$[13,24]→P$[85,96];12→J;gsb "HEADER"
351: for N=0 to Z;N→I
352: prnd(C[N,11]10,0)→O
353: prnd(C[N,7]10,0)→T
354: prnd(C[N,13]/100,0)→S
355: prnd(C[N,14]*16/90,0)→A
356: if A=0;64→A
357: if S=0;0→A
358: if S>99;S-100→S;I+80→I
359: gsb "ARRAY-12"
360: next N
361: gsb "FOOT"
362: spc 5
363: trk 0;ldf 13
364: end
*11177
```

(2) Ballistic 2

TRACK 0 FILE 13

```
0: sfg 3;csp "BALLISTIC TYPE 2";gsb "BEE"
1: gto +70
2: "BEE":beep;wait 150;beep;wait 150;beep;wait 3000;ret
3: "FOOT":fmt 2,"9",/
4: if flg4 and flg8;rar 2;wtc 2,2;wrt 2,2;wtc 2,0
5: fmt 6,2
6: if flg4 and flg7;rar 0;wtc 2,2;wrt 2,3,2,0;wtc 2,0
7: if flg4;wtc 2,2;for N=1 to 30;wtb 2,0;next N;wtc 2,0
8: ret
9: "A→B":fmt 8,2;rar 0
10: for C=1 to 8
11: for K=1 to 2;rcs(A$[K],A$[C,C])→B;if B;gto K;gto +2
12: next K
13: if B=1 or B=3 or B=5 or B=9;gto 1,2,5,6;gto +5
14: if flg1 and flg5;gto 1;gto +4
15: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;gto 1,6;gto +3
16: if flg2 and flg6;gto 2;gto +2
17: sfg 6;gto 2,5;wtc 2,2;wrt 2,31;wtc 2,0
18: wtc 2,2;wrt 2,B-1;wtc 2,0
19: next C
20: wtc 2,2;wrt 2,2,8;wtc 2,0
21: ret
22: "R→P":√(XX+YY)→R[1]
23: atn(Y/(X+1e-30*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
24: A→R[2]
25: if A<0;360+A→R[2]
26: ret
27: "DEAFEN":gto 1,2,5,6;rar 2
28: if flg4 and flg8;fmt 3,c12;wtc 2,2;wrt 2,3,A$[I,12];wtc 2,0
29: if flg4 and flg7;12→I;gsb "A→B"
30: csp A$[I,12];rrt A$[1,12]
31: B$[85,99]→A$[1,J]
32: if flg9;gto +4
33: if flg4 and flg8;wtc 2,2;wrt 2,3,A$[I,J];wtc 2,0
34: if flg4 and flg7;12→I;gsb "A→B"
35: gto +3
36: if flg4 and flg8;fmt 3,c15;wtc 2,2;wrt 2,3,A$[I,J];wtc 2,0
37: if flg4 and flg7;15→B;gsb "A→B"
38: csp A$[I,J];rrt A$[I,J]
39: ret
40: "DELETE":gto 3;csp "MESSAGE BEING DELETED";wait 1000
41: ret
42: "P→R":R[1]*cos(R[2])→X;R[1]*sin(R[2])→Y
43: ret
44: "STRING":fxd 0;str(N)→E$[1,5]
45: if N<1000;B$[2,4]→E$[3,5];"0"→E$[2,2]
46: if N<100;B$[2,4]→E$[3,5];"0"→E$[2,2]
47: if N<10;B$[2,4]→E$[3,5];"0"→E$[2,2]
48: if N<1;B$[2,4]→E$[3,5];"0"→E$[2,2]
49: ret
```

*23874

```
50: "ARRAY":I→M;gsb "STRINC"
51: B$[4,5]→A$[1,2];A→M;gsb "STRINC"
52: B$[3,5]→A$[3,5];S→M;gsb "STRINC"
53: B$[3,5]→A$[6,8];T→M;gsb "STRINC"
54: B$[2,5]→A$[9,12];O→M;gsb "STRINC"
55: B$[2,5]→A$[13,16]
56: if flg4 and flg8;rar 2;fmt 4,c16;wtc 2,2;wrt 2.4,A$[1,16];wtc 2,0
57: if flg4 and flg7;16→P;gsb "A→P"
58: dsp A$[1,16];prt A$[1,16]
59: ret
60: "ARRAY-12":I→M;gsb "STRINC"
61: B$[4,5]→A$[1,2];A→M;gsb "STRINC"
62: B$[4,5]→A$[3,4];S→M;gsb "STRINC"
63: B$[4,5]→A$[5,6];T→M;gsb "STRINC"
64: B$[3,5]→A$[7,9];O→M;gsb "STRINC"
65: B$[3,5]→A$[10,12]
66: if flg4 and flg8;rar 2;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
67: if flg4 and flg7;12→P;gsb "A→P"
68: dsp A$[1,12];prt A$[1,12]
69: ret
70: "T2":
71: E+1→E;for N=1 to Z;rNC[E,6]+C[N,8]→C[N,8];next N
72: ret
73: "D2":
74: E+1→E;for N=1 to Z;rNC[E,10]+C[N,12]→C[N,12];next N
75: ret
76: "PW":
77: C[I,2]→B[1];C[I,5]→B[2];1+I→I;gsb "P→B"
78: for N=1 to Z;XrN+C[N,13]→C[N,13];YrN+C[N,14]→C[N,14];next N
79: ret
80: 100C[0,6]→C[0,8]
81: 100→r1;63→r2;37→r3;25→r4;20→r5;13→r6;10→r7;9→r8;7→r9;5→r10→r11→r13→r1
82: 4→r12;5→r15;0→E;gsb "T2"
83: 0→r1;37→r2;37→r3;30→r4;24→r5;19→r6;14→r7;10→r8;9→r9;8→r10
84: 6→r11→r12→r13→r14→r15;gsb "T2"
85: 0→r2;26→r3;35→r4;30→r5;24→r6;20→r7;17→r8;14→r9;12→r10;10→r11→r12→r13→
86: 10→r15;gsb "T2"
87: 0→r3;10→r4;18→r5→r6;16→r7;15→r8;13→r9;10→r10;8→r12
88: 9→r11→r13→r14→r15;gsb "T2"
89: 0→r4;8→r5;14→r6→r7;13→r8;12→r9;10→r10;8→r11→r13→r14→r15;gsb "T2"
90: 0→r5;12→r6;19→r7;20→r8;19→r9;17→r10;15→r11;14→r12;16→r13→r14→r15;gsb
91: 0→r6;7→r7;12→r8;15→r9;14→r10;13→r11→r12;12→r13→r14→r15;gsb "T2"
92: 0→r7;4→r8;8→r9;10→r10;12→r11;11→r12;13→r13→r14→r15;gsb "T2"
93: 0→r8;3→r9;8→r10;10→r11→r12;11→r13→r14→r15;gsb "T2"
94: 0→r9;6→r10;12→r11;16→r12;10→r13→r14→r15;gsb "T2"
95: 100C[0,10]→C[0,12]
96: 100→r1;63→r2;37→r3;25→r4;20→r5;13→r6;10→r7;9→r8;7→r9;5→r10;4→r11→r12
97: 3→r13→r14;2→r15;0→E;gsb "D2"
98: 0→r1;37→r2;37→r3;30→r4;24→r5;19→r6;14→r7;10→r8;9→r9;8→r10;6→r11→r12
99: 5→r13→r14→r15;gsb "D2"
*32459
```

```
100: 0→r2;26→r3;35→r4;30→r5;24→r6;20→r7;17→r8;14→r9;12→r10;10→r11;9→r12
101: 8→r13;6→r14→r15;gsb "D2"
102: 0→r3;10→r4;18→r5→r6;16→r7;15→r8;13→r9;10→r10;8→r11→r12
103: 7→r14→r15;gsb "D2"
104: 0→r4;8→r5;14→r6→r7;13→r8;12→r9;6→r13;5→r15;gsb "D2"
105: 0→r5;12→r6;19→r7;20→r8;19→r9;17→r10;15→r11;13→r12;12→r13;11→r14→r15
106: gsb "D2"
107: 0→r6;7→r7;12→r8;15→r9;14→r10;13→r11;12→r12;11→r13;10→r14→r15;gsb "D2"
108: 0→r7;4→r8;8→r9;10→r10→r11→r12→r13;9→r14;8→r15;gsb "D2"
109: 0→r8;3→r9;8→r10;10→r11;8→r12→r13→r14→r15;gsb "D2"
110: 0→r9;6→r10;12→r11;13→r12→r13→r14→r15;gsb "D2"
111: 0→r10;4→r11;7→r12;9→r13;10→r14→r15;gsb "D2"
112: 0→r11;2→r12;5→r13;6→r14;7→r15;gsb "D2"
113: 0→r12;2→r13;4→r14;5→r15;gsb "D2"
114: 0→r13;1→r14;3→r15;gsb "D2"
115: for N=1 to Z;0→C[N,13]→C[N,14];next N
116: 1→I
117: 100→r1;50→r2;29→r3;18→r4;13→r5;8→r6;7→r7;4→r8→r9;3→r10→r12;1→r15
118: 2→r11→r13→r14;gsb "WW"
119: 0→r1;50→r2;33→r3;23→r4;18→r5;12→r6;8→r7→r8;6→r9;3→r15
120: 4→r10→r11→r12→r13→r14;gsb "WW"
121: 0→r2;38→r3;39→r4;31→r5;22→r6;16→r7;13→r8;11→r9;8→r10;6→r11;7→r12
122: 5→r13→r14→r15;gsb "WW"
123: 0→r3;20→r4;27→r5;20→r6;15→r7;12→r8;10→r9;7→r11;6→r13→r14;4→r15;gsb "WW"
124: 0→r4;11→r5;19→r6;16→r7;13→r8;6→r11;4→r14;5→r15;gsb "WW"
125: 0→r5;19→r6;27→r7;24→r8;21→r9;16→r10;13→r11;12→r12;11→r13;9→r14→r15
126: gsb "WW"
127: 0→r6;11→r7;18→r8;20→r9;15→r10;11→r12;10→r13;gsb "WW"
128: 0→r7;8→r8;12→r9;14→r10;12→r11;10→r12;9→r13;8→r15;gsb "WW"
129: 0→r8;6→r9;13→r10;11→r11;8→r12→r13→r14;7→r15;gsb "WW"
130: 0→r9;11→r10;18→r11;15→r12;14→r13;13→r14;12→r15;gsb "WW"
131: 0→r10;8→r11;10→r12;11→r13→r14;10→r15;gsb "WW"
132: 0→r11;6→r12;9→r13→r14→r15;gsb "WW"
133: 0→r12;5→r13;6→r14;8→r15;gsb "WW"
134: 0→r13;5→r14;6→r15;gsb "WW"
135: 0→r14;4→r15;gsb "WW"
136: for N=1 to Z;C[N,13]→X;C[N,14]→Y;gsb "P→F"
137: F[1]→C[N,13];F[2]→C[N,14];next N
138: H[-2,5]→C[0,14];H[-2,6]100→C[0,13]
139: if not flg3;gto +14
140: P$[37,48]→A$[1,12];P$[13,24]→F$[85,96];12→J;gsb "HEADER"
141: for N=0 to Z;N→I
142: prnd(C[N,12]10,0)→Q
143: prnd(C[N,8]10,0)→T
144: prnd(C[N,13]/100,0)→S
145: prnd(C[N,14]*16/90,0)→A
146: if A=0;64→A
147: if S=0;0→A
148: if S>99;S-100→S;I+80→I
149: gsb "ARRAY-12"
*10299
```

```
150: next N
151: gsb "FOOT"
152: spc 5
153: if P$[100,100]="N";trk 0;ldf 15
154: dim T[4];"////////"→C$[6,13]
155: 6.11*10^(7.5F[0,1]/(237.3+F[0,1]))→V
156: if F[0,4]<850;gto +3
157: if F[0,4]<1000 and F[0,4]>=850;"N"→C$[6,6];gto +12
158: if F[0,4]>=1000;"N///N"→C$[6,10];gto +16
159: F[0,1]+.07(850-F[0,4])+273.16→T[2]
160: T[2](1+.151212V/850)→T[4]
161: 6.11*10^(7.5(T[2]-273.16)/(237.3+T[2]-273.16))→V
162: T[2]+(.07)150→T[1]
163: T[1](1+.151212V/1000)→T[3]
164: 10val(P$[19,21])-33.721(T[4]+F[0,3])*log(850/F[0,4])→M;gsb "STFINC"
165: B$[3,5]→C$[6,8]
166: M-33.721(T[3]+T[4])*log(1000/850)→M;gsb "STFINC"
167: if M<0;M+500→M;str(M)→C$[10,13];C$[11,13]→C$[10,12];gto +8
168: if M>0;gsb "STFINC";B$[3,5]→C$[10,12];gto +6
169: F[0,1]+.07(1000-F[0,4])+273.16→T[1]
170: T[1](1+.151212V/1000)→T[3]
171: 10val(P$[19,21])-33.721(T[3]+F[0,3])*log(1000/F[0,4])→M;gsb "STFINC"
172: if M<0;M+500→M;str(M)→C$[10,13];C$[11,13]→C$[10,12];gto +2
173: B$[3,5]→C$[10,12]
174: trk 0;rcf 9,B$,C$,C$,L$,M$,C$,D$
175: trk 0;ldr 14
176: end
*15339
```

(3) AWS Output First Transmission

TRACK 0 FILE 14

```
0: dim A$[20],P$[15],C[1],B[12]
1: dim P$[102],Q$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
2: dim B[-2:45,1:6]
3: dim F[-1:60,1:6],Y[2]
4: dim M[3,45],C$[16]
5: dim V$[40],Z$[16],A[6]
6: dim I$[5],E$[10]
7: dim H$[2,32];time 500
8: cfg 4;if ios2;sfg 4
9: sfg 3;dsp "USAF AWS MESSAGE";gsb "BEP"
10: trk 0;ldf 9,P$,Q$,O$,L$,N$,C$,D$
11: trk 0;ldf 10,F[*],Y[*]
12: trk 0;ldf 11,B[*]
13: if P$[101,101]="Y";sfg 7;cfg 8
14: if P$[101,101]="N";cfg 7;sfg 8
15: " 5 9 #,. )4&80:;3 $? 6!/-2' 71( "→H$[1]
16: " 7 O HNM LPCIPCVFZDBSYFXAWJ UQK "→H$[2]
17: char(0)→H$[1,1,1];char(10)→H$[1,9,9];char(7)→H$[1,21,21]
18: char(13)→H$[1,3,3]
19: gsb "INTRT"
20: gto +125
21: "BEP":beep;wait 150;beep;wait 150;beep;wait 3000;ret
22: "INTRT":for N=1 to Y[1];if F[N,3]#0;gto +7
23: for K=N to Y[1];if F[N,3]=0;next K
24: (F[K,5]-F[N,5])/(F[K,5]-F[N-1,5])→C
25: F[K,1]-C(F[K,1]-F[N-1,1])→F[N,1]
26: F[K,2]-C(F[K,2]-F[N-1,2])→F[N,2]
27: F[K,3]-C(F[K,3]-F[N-1,3])→F[N,3]
28: F[K,6]-C(F[K,6]-F[N-1,6])→F[N,6]
29: next N
30: ret
31: "A→B":fmt 1,z;par 0
32: for C=1 to r14
33: for K=1 to 2;pos(H$[K],C$[C,C])→J;if J;sfg K;gto +2
34: next K
35: if J=1 or J=3 or J=5 or J=9;cfg 1,2,5,6;gto +5
36: if flg1 and flg5;cfg 1;gto +4
37: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
38: if flg2 and flg6;cfg 2;gto +2
39: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
40: wtc 2,2;wrt 2,J-1;wtc 2,0
41: next C
42: ret
43: "FOOT":if flg4 and flg7;par 0;fmt 2,1,z;wtc 2,2;wrt 2.2,15;wtc 2,0
44: fmt 2,z,c1
45: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2,char(59);wtc 2,0
46: dsp ";";prt ";"
47: 10→C[1]
48: ret
49: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000;ret
*29139
```

```
50: "ARRAY 15":
51: "        "→C$[1,6]
52: A$[1,5]→C$[2,6];gsb "ARRAY 6"
53: A$[6,10]→C$[2,6];gsb "ARRAY 6"
54: A$[11,15]→C$[2,6];gsb "ARRAY 6"
55: dsp A$[1,15];prt A$[1,15]
56: ret
57: "ARRAY 10":
58: "        "→C$[1,6]
59: A$[1,5]→C$[2,6];gsb "ARRAY 6"
60: A$[6,10]→C$[2,6];gsb "ARRAY 6"
61: dsp A$[1,10];prt A$[1,10]
62: ret
63: "ARRAY 5":
64: "        "→C$[1,6]
65: A$[1,5]→C$[2,6];gsb "ARRAY 6"
66: dsp A$[1,5];prt A$[1,5]
67: ret
68: "ARRAY 6":
69: if C$[1,6]="        ";ret
70: C[1]+1→C[1];if C[1]<10;gto +7
71: if flg4 and flg8;par 2;fmt 4,z,b;wtc 2,2;wrt 2.4,14],10;wtc 2,0
72: if flg4 and flg7;fmt 4,b,z;par 0;wtc 2,2;wrt 2.4,2,8;wtc 2,0
73: 0→C[1];C$[2,6]→C$[1,5]
74: if flg4 and flg7;5→r14;gsb "A→B"
75: if flg4 and flg8;par 2;fmt 4,z,c5;wtc 2,2;wrt 2.4,C$[1,5];wtc 2,0
76: gto +3
77: if flg4 and flg8;par 2;fmt 4,z,c6;wtc 2,2;wrt 2.4,C$[1,6];wtc 2,0
78: if flg4 and flg7;6→r14;gsb "A→B"
79: "        "→C$[1,6]
80: ret
81: "STRING":if Q=0;"000"→A$[3,5]
82: if Q>999;str(Q)→A$[6,10];A$[8,10]→A$[3,5]
83: if Q>99 and Q<1000;str(Q)→A$[6,9];A$[7,9]→A$[3,5]
84: if Q>9 and Q<100;str(Q)→A$[3,5];"0"→A$[3,3]
85: if Q<10;str(Q)→A$[4,5];"00"→A$[3,4]
86: ret
87: "TEMP":prnd(F[F,1],-1)→T;frc(T)→Q
88: if sgn(T)=1 or sgn(T)=0;if Q=.1 or Q=.3 or Q=.5 or Q=.7 or Q=.9;T-.1→T
89: if sgn(T)=-1;if Q=.2 or Q=.4 or Q=.6 or Q=.8 or Q=0;T-.1→T
90: abs(T)→T
91: "000"→A$[6,8];str(10T)→C$
92: if T>0 and T<1;C$[2,2]→A$[8,8]
93: if T>.9 and T<10;C$[2,3]→A$[7,8]
94: if T>9.9;C$[2,4]→A$[6,8]
95: ret
96: "DP":if prnd(F[F,1],-1)<-40;"//"→A$[9,10];ret
97: if F[F,2]=0;2→F[F,2]
98: (F[F,2]/100)6.11tr^(7.5F[F,1]/(237.3+F[F,1]))→F
99: (237.3log(F)-186.527)/(8.286-log(F))→F
*8238
```

```
100: prnd(F[F,1]-E,-1)→E
101: "00"→A$[9,10]
102: if E>0 and E<1;str(prnd(10E,0))→C$;C$[2,2]→A$[10,10]
103: if E>.9 and E<5;str(prnd(10E,0))→C$;C$[2,3]→A$[9,10]
104: if E>4.9;str(prnd(50+E,0))→C$;C$[2,3]→A$[9,10]
105: ret
106: "ANGLES":for N=2 to 44;if O>M[1,N];next N
107: if M[3,N]=0 and M[2,N]=0;"      "→A$[11,15];ret
108: M[1,N-1]→r1;M[1,N]→r2;M[3,N-1]→r3;M[3,N]→r4;M[2,N-1]→r5;M[2,N]→r6
109: r4-((r2-O)/(r2-r1))(r4-r3)→S
110: if r5-r6>=180;r6+360→r6
111: if r6-r5>=180;r5+360→r5
112: r6-(r2-O)(r6-r5)/(r2-r1)→A
113: if S<0;A+180→A;abs(S)→S
114: if A>=360;A-360→A
115: if A<0;A+360→A
116: prnd(A/5,0)*5→A;prnd(S,0)→S;S+100A→A
117: str(A)→C$;C$[2,6]→A$[11,15]
118: if A<10000;"0"→A$[11,11];C$[2,5]→A$[12,15]
119: if A<1000;"00"→A$[11,12];C$[2,4]→A$[13,15]
120: if A<100;"000"→A$[11,13];C$[2,3]→A$[14,15]
121: if A<10;"0000"→A$[11,14];C$[2,2]→A$[15,15]
122: if A=0;"000"→A$[11,13]
123: ret
124: "HEIGHT & TEMP & DEW POINT DEPRESSION":
125: prnd(F[F,6]+val(P$[19,21])*10,0)→O
126: str(O)→C$;"000"→A$[3,5]
127: if O<=500;gto +5
128: if O>0 and O<10;C$[2,2]→A$[5,5]
129: if O>9 and O<100;C$[2,3]→A$[4,5]
130: if O>99 and O<1000;C$[2,4]→A$[3,5]
131: if O>999;C$[3,5]→A$[3,5];gto +3
132: if O>999 and O<10000;C$[2,4]→A$[3,5]
133: if O>9999;C$[3,5]→A$[3,5]
134: gsb "TEMP"
135: gsb "DP"
136: prnd(F[F,6]+val(P$[19,21])*10,0)→O
137: gsb "ANGLES"
138: ret
139: "R→P":√(XX+YY)→R
140: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
141: if A<0;360+A→A
142: ret
143: "P→R":R*cos(A)→X;R*sin(A)→Y
144: ret
145: if not flg3;gto +207
146: 10→C[1]
147: "MPTW"→A$[1,4];P$[6,9]→A$[5,8]
148: D$[4,5]→A$[9,10];P$[10,12]→A$[11,13];D$[6,7]→A$[14,15]
149: gsb "ARRAY 15"
*23470
```

```
150: P$[13,14]→A$[1,2];P$[19,21]→A$[3,5];gsb "ARRAY 5"
151: 10→C[1]
152: gto +8
153: "SUBHEAD":                "→A$[6,15]
154: val(D$[1,2])→Z;str(prnd(Z+50,0))→A$[6,8]
155: A$[7,8]→A$[6,7]
156: str(prnd(val(C$[4,7]).01,0))→A$[8,10];A$[9,10]→A$[8,9]
157: if val(A$[8,9])<10;A$[8,8]→A$[9,9];"0"→A$[8,8]
158: C$[1,5]→A$[11,15]
159: ret
160: 10val(P$[19,21])→M[1,1];H[-2,5]→M[2,1];H[-2,6]→M[3,1]
161: H[-1,5]→A;H[-1,6]→B;gsb "R→P"
162: X→r1;Y→r2
163: .0001→H[0,4]
164: for P=1 to 45;if H[P,4]=0;gto +11
165: P-1→J
166: 6367650→S
167: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→r3
168: (Scos(H[P,6])/(S+H[P,1]))r3→R
169: H[P,5]→A;gsb "R→P"
170: r1→r4;X→r1;r4-r1→X;r2→r5;Y→r2;r5-r2→Y;gsb "R→P"
171: (P/(H[P,4]-H[J,4])).03238→S→M[3,P+1]
172: A→M[2,P+1]
173: H[J,1]+(H[P,1]-H[J,1])/2+10val(P$[19,21])→H→M[1,P+1]
174: next P
175: "10000850070005000400030002500200015001 00"→W$
176: src 1;" TTAA"→A$[1,5];gsb "SUBHEAD"
177: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
178: M[1,N]→B[5]
179: for N=2 to Y[1];if F[N,6]<B[5];next N
180: "/"→A$[10,10]
181: for W=9 to 0 by -1
182: if F[N-1,4]<=val(W$[W4+1,W4+4]);W$[W4+2,W4+2]→A$[10,10];gto +2
183: next W
184: gsb "ARRAY 15"
185: "////////////////"→A$
186: 0→E;gsb "REICHT & TEMP & DEW POINT DEPRESSION"
187: "99"→A$[1,2];str(F[0,4])→C$;C$[2,4]→A$[3,5]
188: if F[0,4]>1000;C$[3,5]→A$[3,5]
189: H[-2,5]→A;H[-2,6]→S;prnd(A/5,0)*5→A
190: prnd(S,0)→S;S+100A→A;str(A)→A$[11];A$[12,16]→A$[11,15]
191: if A<10000;A$[11,14]→A$[12,15];"0"→A$[11,11]
192: if A<1000;A$[11,14]→A$[12,15];"0"→A$[11,11]
193: if A=0;"000"→A$[11,13]
194: A$[11,15]→Z$[1,5]
195: gsb "ARRAY 15"
196: 0→W
197: for F=1 to Y[1]
198: val(W$[W4+1,W4+4])→P
199: "////////////////"→A$
*23417
```

```
200: if P=1000 and C$[10,10]#"N";C$[10,12]→A$[3,5];gto +5
201: if P=850 and C$[6,6]#"N";C$[6,8]→A$[3,5];gto +4
202: if P<F[F,4] and P#F[F,4];gto +7
203: if P=F[-1,3] or P=F[-1,4] or P=F[-1,5] or P=F[-1,6];gto +2
204: gsb "HEIGHT & TEMP & DEW POINT DEPRESSION"
205: W$[W4+2,W4+3]→A$[1,2]
206: gsb "ARRAY 15"
207: if P#100;W+1→W
208: if P=100;A$[3,5]→Z$[6,8];gto +2
209: next F
210: if F[Y[1],4]>200;gto +8
211: for L=1 to Y[1];if F[L,4]>500;next L
212: L+1→N
213: if N>Y[1];gto +5
214: prnd((F[L,3]-F[N,3])/.001(F[N,6]-F[L,6]),-1)→C
215: if C<=2 and F[N,6]-F[L,6]>=2000;gto +4
216: if C<=2 and F[N,6]-F[L,6]<2000;N+1→N;gto -3
217: next L
218: gto +13
219: F[L,4]→O;F[L,6]→P;L→F
220: "/////////////////"→A$
221: gsb "TEMP"
222: gsb "DP"
223: W+10val(P$[19,21])→O;gsb "ANGLES"
224: "88"→A$[1,2];str(O)→C$;C$[2,4]→A$[3,5]
225: if val(A$[3,5])<100;gto +3
226: gsb "ARRAY 15"
227: gto +5
228: val(A$[1,5])→A[1];val(A$[11,15])→A[3]
229: if A$[9,10]="//";val(A$[6,8])→A[2]
230: if A$[9,10]#"//";val(A$[6,10])→A[2];gto +2
231: "88999"→A$;gsb "ARRAY 5"
232: if F[Y[1],4]>500;gto +27
233: M[3,1]→V
234: for N=2 to 46;if B[N-1,4]=0;gto +3
235: if M[3,N]>V;M[3,N]→V;M[2,N]→O;N→C
236: next N
237: if V<60;gto +21
238: 1000/(M[1,C]-M[1,C-1])→r13;1000/(M[1,C+1]-M[1,C])→r12
239: abs(V-M[3,C-1])r13→r13;abs(V-M[3,C+1])r12→r12
240: for L=1 to 60;if F[L,6]<=M[1,C] and F[L+1,6]>M[1,C];gto +2
241: next L
242: if F[L,4]>500;gto +16
243: F[L,4]+(M[1,C]-F[L,6])(F[L+1,4]-F[L,4])/(F[L+1,6]-F[L,6])→P
244: "77"→A$[1,2];P→Q;if P<100;10P→Q
245: gsb "STRING"
246: prnd(O/5,0)5→O;prnd(V,0)→V;1000+V→V
247: str(V)→C$;C$[2,6]→A$[6,10];"4"→A$[11,11]
248: if r13<1;100+prnd(r12,0)→r13;str(r13)→C$;C$[2,5]→A$[12,15]
249: 100prnd(r13,0)+prnd(r12,0)→r13;str(r13)→C$;C$[2,5]→A$[12,15]
*6390
```

```
250: if r13<1000;"0"→A$[12,12];C$[2,4]→A$[13,15]
251: if B[C+1,4]=0;"//"→A$[14,15]
252: if F<100;gto +3
253: gsb "ARRAY 15"
254: gto +5
255: val(A$[1,5])→A[4];val(A$[6,10])→A[5]
256: if A$[14,15]="//";val(A$[11,13])→A[6]
257: if A$[14,15]#"//";val(A$[6,10])→A[6]
258: "77999"→A$;gsb "ARRAY 5"
259: gsb "FOOT"
260: spc 1;" TTBB"→A$[1,5];gsb "SUBHEAD"
261: "/"→A$[10,10];gsb "ARRAY 15"
262: 1→B[11];0→W;F[0,1]→B[3]→B[4];F[0,2]→B[5]→B[6]
263: for M=1 to Y[1]
264: if prnd(F[M,1],-1)>prnd(B[3],-1);F[M,1]→B[3];M→B[7]
265: if prnd(F[M,1],-1)<prnd(B[4],-1);F[M,1]→B[4];M→B[8]
266: if prnd(F[M,2],0)>prnd(B[5],0);F[M,2]→B[5];M→B[9]
267: if prnd(F[M,2],0)<prnd(B[6],0) and F[M,2]#0;F[M,2]→B[6];M→B[10]
268: if F[M,2]=0 and B[11]#0;0→B[11];M-1→B[12]
269: next M
270: gto +8
271: "BOUND":Y[1]→B[2]
272: if B[7]>B[1] and B[7]<B[2];B[7]→B[2]
273: if B[8]>B[1] and B[8]<B[2];B[8]→B[2]
274: if B[9]>B[1] and B[9]<B[2];B[9]→B[2]
275: if B[10]>B[1] and B[10]<B[2];B[10]→B[2]
276: if B[12]>B[1] and B[12]<B[2];B[12]→B[2]
277: ret
278: 0→B[1]→F;gto +20
279: 0→B[3]→A→B[4]
280: for M=B[1]+1 to B[2]
281: (F[M,6]-F[B[1],6])/(F[B[2],6]-F[B[1],6])→C
282: F[M,1]-F[B[1],1]-C(F[B[2],1]-F[B[1],1])→A
283: if prnd(F[M,1],-1)=prnd(F[B[7],1],-1);M→B[2];gto +5
284: if prnd(F[M,1],-1)=prnd(F[B[8],1],-1);M→B[2];gto +4
285: if abs(A)>1 and F[M,4]>=300 and abs(A)>B[4];M→B[3];abs(A)→B[4]
286: if abs(A)>2 and F[M,4]<300 and abs(A)>B[4];M→B[3];abs(A)→B[4]
287: next M
288: if B[3]>B[1] and B[3]<B[2];B[3]→B[2];gto -9
289: 0→B[3]→A;if B[1]>B[12];gto +8
290: for M=B[1]+1 to B[2]
291: (F[M,6]-F[B[1],6])/(F[B[2],6]-F[B[1],6])→C
292: F[M,2]-F[B[1],2]-C(F[B[2],2]-F[B[1],2])→A
293: if prnd(F[M,2],0)=prnd(F[B[9],2],0);M→B[2];gto +4
294: if prnd(F[M,2],0)=prnd(F[B[10],2],0);M→B[2];gto +3
295: if abs(A)>10 and F[M,2]>20;M→B[2];gto +2
296: next M
297: B[2]→F
298: if F[F,4]<100;gto +13
299: "//////////"→A$[1,10];if D>9;1→D
*4276
```

```
300: str(prnd(D11,0))→A$[1,3];A$[2,3]→A$[1,2];"/"→A$[3,3]
301: if L=0;"00"→A$[1,2]
302: prnd(F[F,4],0)→Q
303: gsb "STRING"
304: gsb "TEMP"
305: gsb "DP"
306: 1+D→D
307: gsb "ARRAY 10"
308: if F=Y[1];gto +3
309: F→F[1];gsb "BOUND"
310: gto -31
311: gsb "FOOT"
312: spc 1;" PPBP"→A$[1,5];gsb "SUBHEAD"
313: "0"→A$[10,10];gsb "ARRAY 15"
314: "010203040607080912141620253035507 0"→W$[1,34]
315: int(val(P$[19,21])*.03281+1)→r10
316: int((F[Y[1],6]/10+val(P$[19,21]))*.03281)→r9
317: if F[Y[1],4]>100;gto +2
318: int((val(Z$[6,8])+val(P$[19,21])+1000)*.03281)→r9
319: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
320: if int(M[1,N]*.003281)<r9;int(M[1,N]*.003281)→r9
321: "900//"→I$[1,5];2→D
322: "                    "→F$[1,15];Z$[1,5]→F$[1,5]
323: for W=0 to 16
324: val(W$[W2+1,W2+2])→Q
325: if r10>Q;next W
326: if Q>r9;gto +13
327: Q*304.8→Q
328: gsb "ANGLES"
329: if D=1;"9////"→I$[1,5];"                    "→F$[1,15]
330: if D=1;W$[W2+1,W2+2]→I$[2,3];A$[11,15]→F$[1,5]
331: if D=2;W$[W2+2,W2+2]→I$[4,4];A$[11,15]→F$[6,10]
332: if D=3;W$[W2+2,W2+2]→I$[5,5];A$[11,15]→F$[11,15]
333: if D#3 and W$[W2+1,W2+1]=W$[W2+3,W2+3];D+1→D;next W
334: str(r9)→B$;if r9>=10 and B$[2,2]=W$[W2+1,W2+1];D+1→D;next W
335: if r9<10 and W$[W2+1,W2+1]="0";D+1→D;next W
336: I$[1,5]→A$[1,5];gsb "ARRAY 5"
337: F$[1,15]→A$[1,15];gsb "ARRAY 15"
338: 1→D;next W
339: r9*304.8→Q;gsb "ANGLES"
340: if r9=val(W$[W2-1,W2]) and D=1;gto +10
341: if r9=val(W$[W2-1,W2]) and D#1;gto +7
342: str(r9)→B$;if r9<10;B$[2,2]→B$[3,3];"0"→B$[2,2]
343: if W$[W2-1,W2-1]#B$[2,2];1→D
344: if D=1;"9////"→I$[1,5];"                    "→F$[1,15]
345: if D=1;B$[2,3]→I$[2,3];A$[11,15]→F$[1,5]
346: if D=2;B$[3,3]→I$[4,4];A$[11,15]→F$[6,10]
347: if D=3;B$[3,3]→I$[5,5];A$[11,15]→F$[11,15]
348: I$[1,5]→A$[1,5];gsb "ARRAY 5"
349: F$[1,15]→A$[1,15];gsb "ARRAY 15"
*12315
```

```
350: gsb "FOOT"
351: if F[Y[1],4]<100;trk 0;ldf 16
352: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
353: trk 0;ldf 15
354: end
*22143
```

(4) Last Flight Update

TRACK 0 FILE 15

```
0: 2→A;beep;dsp "NEED ""LAST FLIGHT"" UPDATING?";ent "",A
1: if A#-2tn^10 and A#-1tn^10;gto -1
2: if A=-2tn^10;gto +8
3: dsp """LAST FLIGHT"" BEING UPDATED";gsb "BBB"
4: trk 0;ldf 9,P$,O$,O$,L$,N$,C$,D$
5: trk 0;ldf 10,F[*],Y[*]
6: trk 0;ldf 11,H[*]
7: trk 1;rcf 5,P$,O$,O$,L$,N$,C$,D$
8: trk 1;rcf 6,F[*],Y[*]
9: trk 1;rcf 7,H[*]
10: if ios2;dsp "CONSERVE ENERGY  TURN-OFF REMEX";gsb "BBB"
11: dsp "PROCESSING FLIGHT CONTINUATION";gsb "BBB"
12: spc 5;trk 0;ldp 6
13: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
14: end
*2151
```

(5) AWS Output Second Transmission.

TRACK 0 FILE 16

```
0: dsp "USAF AWS MESSAGE"
1: gto +113
2: "A→B":fmt b,z;par 0
3: for Q=1 to r14
4: for K=1 to 2;pos(B$[K],C$[Q,Q])→J;if J;sfg K;gto +2
5: next K
6: if J=1 or J=3 or J=5 or J=9;cfg 1,2,5,6;gto +5
7: if flg1 and flg5;cfg 1;gto +4
8: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
9: if flg2 and flg6;cfg 2;gto +2
10: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
11: wtc 2,2;wrt 2,J-1;wtc 2,0
12: next Q
13: ret
14: "ECOT":if flg4 and flg7;par 0;fmt 2,b,z;wtc 2,2;wrt 2.2,15;wtc 2,0
15: fmt 2,z,c1
16: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2,char(59);wtc 2,0
17: dsp ";";prt ";"
18: 10→C[1]
19: ret
20: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000;ret
21: "ARRAY 15":
22: "          "→C$[1,6]
23: A$[1,5]→C$[2,6];gsb "ARRAY 6"
24: A$[6,10]→C$[2,6];gsb "ARRAY 6"
25: A$[11,15]→C$[2,6];gsb "ARRAY 6"
26: dsp A$[1,15];prt A$[1,15]
27: ret
28: "ARRAY 10":
29: "          "→C$[1,6]
30: A$[1,5]→C$[2,6];gsb "ARRAY 6"
31: A$[6,10]→C$[2,6];gsb "ARRAY 6"
32: dsp A$[1,10];prt A$[1,10]
33: ret
34: "ARRAY 5":
35: "          "→C$[1,6]
36: A$[1,5]→C$[2,6];gsb "ARRAY 6"
37: dsp A$[1,5];prt A$[1,5]
38: ret
39: "ARRAY 6":
40: if C$[1,6]="          ";ret
41: C[1]+1→C[1];if C[1]<10;gto +8
42: par 2
43: if flg4 and flg8;fmt 4,z,b;wtc 2,2;wrt 2.4,141,10;wtc 2,0
44: if flg4 and flg7;fmt 4,b,z;par 0;wtc 2,2;wrt 2.4,2,8;wtc 2,0
45: 0→C[1];C$[2,6]→C$[1,5]
46: if flg4 and flg7;5→r14;gsb "A→B"
47: if flg4 and flg8;par 2;fmt 4,z,c5;wtc 2,2;wrt 2.4,C$[1,5];wtc 2,0
48: gto +3
49: if flg4 and flg8;par 2;fmt 4,z,c6;wtc 2,2;wrt 2.4,C$[1,6];wtc 2,0
*26909
```

```
50: if flg4 and flg7;6→r14;gsb "A→B"
51: "        "→C$[1,6]
52: ret
53: "STRING":if Q<1000 and Q>99;str(Q)→A$[3,6];A$[4,6]→A$[3,5]
54: if Q<100 and Q>9;str(Q)→A$[3,5];"0"→A$[3,3]
55: ret
56: "TEMP":prnd(F[F,1],-1)→T;frc(T)→Q
57: if sgn(T)=1 or sgn(T)=0;if Q=.1 or Q=.3 or Q=.5 or Q=.7 or Q=.9;T-.1→T
58: if sgn(T)=-1;if Q=.2 or Q=.4 or Q=.6 or Q=.8 or Q=0;T-.1→T
59: abs(T)→T
60: "000"→A$[6,8];str(10T)→C$
61: if T>0 and T<1;C$[2,2]→A$[8,8]
62: if T>.9 and T<10;C$[2,3]→A$[7,8]
63: if T>9.9;C$[2,4]→A$[6,8]
64: ret
65: "DP":if prnd(F[F,1],-1)<-40;"//"→A$[9,10];ret
66: if F[F,2]=0;2→F[F,2]
67: (F[F,2]/100)6.11tn^(7.5F[F,1]/(237.3+F[F,1]))→F
68: (237.3log(E)-186.527)/(8.286-log(E))→E
69: prnd(F[F,1]-F,-1)→E
70: "00"→A$[9,10]
71: if E>0 and E<1;str(prnd(10E,0))→C$;C$[2,2]→A$[10,10]
72: if E>.9 and E<5;str(prnd(10E,0))→C$;C$[2,3]→A$[9,10]
73: if E>4.9;str(prnd(50+E,0))→C$;C$[2,3]→A$[9,10]
74: ret
75: "ANGLES":for N=2 to 44;if Q>M[1,N];next N
76: if M[3,N]=0 and M[2,N]=0;"      "→A$[11,15];ret
77: M[1,N-1]→r1;M[1,N]→r2;M[3,N-1]→r3;M[3,N]→r4;M[2,N-1]→r5;M[2,N]→r6
78: r4-((r2-Q)/(r2-r1))(r4-r3)→S
79: if r5-r6>=180;r6+360→r6
80: if r6-r5>=180;r5+360→r5
81: r6-(r2-Q)(r6-r5)/(r2-r1)→A
82: if S<0;A+180→A;abs(S)→S
83: if A>=360;A-360→A
84: if A<0;A+360→A
85: prnd(A/5,0)*5→A;prnd(S,0)→S;S+100A→A
86: str(A)→C$;C$[2,6]→A$[11,15]
87: if A<10000;"0"→A$[11,11];C$[2,5]→A$[12,15]
88: if A<1000;"00"→A$[11,12];C$[2,4]→A$[13,15]
89: if A<100;"000"→A$[11,13];C$[2,3]→A$[14,15]
90: if A<10;"0000"→A$[11,14];C$[2,2]→A$[15,15]
91: if A=0;"000"→A$[11,13]
92: ret
93: "HEIGHT & TEMP & DEW POINT DEPRESSION":
94: prnd(F[F,6]+val(P$[19,21])*10,0)→Q
95: str(Q)→C$;C$[3,5]→A$[3,5]
96: gsb "TEMP"
97: gsb "DP"
98: prnd(F[F,6]+val(P$[19,21])*10,0)→Q
99: gsb "ANGLES"
*8375
```

```
100: ret
101: "R→P":√(XX+YY)→R
102: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
103: if A<0;360+A→A
104: ret
105: "P→R":R*cos(A)→X;R*sin(A)→Y
106: ret
107: "SUBHEAD":"              "→A$[6,15]
108: val(D$[1,2])→Z;str(rnd(Z+50,0))→A$[6,8]
109: A$[7,8]→A$[6,7]
110: str(rnd(val(D$[4,7]).01,0))→A$[8,10];A$[9,10]→A$[8,9]
111: if val(A$[8,9])<10;A$[8,8]→A$[9,9];"0"→A$[8,8]
112: C$[1,5]→A$[11,15]
113: ret
114: "70503020100705030201000000000000000000"→V$
115: spc 2;" TTCC"→A$[1,5];gsb "SUBHEAD"
116: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
117: M[1,N]→B[5]
118: for N=2 to Y[1];if F[N,6]<B[5];next N
119: for W=4 to 0 by -1
120: if F[N-1,4]<=val(W$[W2+1,W2+2]);V$[W2+1,W2+1]→A$[10,10];gto +2
121: next W
122: gsb "ARRAY 15"
123: "////////////////"→A$
124: 0→F;gsb "HEIGHT & TEMP & DEW POINT DEPRESSION"
125: 0→W
126: for F=1 to Y[1]
127: val(W$[W2+1,W2+2])→B
128: "////////////////"→A$
129: if B<F[F,4] and B#F[F,4];gto +6
130: if B=F[-1,3] or B=F[-1,4] or B=F[-1,5] or B=F[-1,6];gto +2
131: gsb "HEIGHT & TEMP & DEW POINT DEPRESSION"
132: W$[W2+1,W2+2]→A$[1,2]
133: gsb "ARRAY 15"
134: if B#1;W+1→W
135: next F
136: if A[1]=0;gto +7
137: str(A[1])→C$;C$[2,6]→A$[1,5];str(A[3])→C$;C$[2,6]→A$[11,15]
138: str(A[2])→C$
139: if A[2]>=1000;C$[2,6]→A$[6,10]
140: if A[2]<1000;C$[2,4]→A$[6,8];"//"→A$[9,10]
141: gsb "ARRAY 15"
142: gto +2
143: "88999"→A$;gsb "ARRAY 5"
144: if A[4]=0;gto +7
145: str(A[4])→C$;C$[2,6]→A$[1,5];str(A[5])→C$;C$[2,6]→A$[6,10]
146: str(A[6])→C$
147: if A[6]<1000;C$[2,4]→A$[11,13];"//"→A$[14,15]
148: if A[6]>=1000;C$[2,6]→A$[11,15]
149: gsb "ARRAY 15"
*28487
```

```
150: gto +2
151: "77999"→A$;gsb "ARRAY 5"
152: gsb "FOOT"
153: spc 1;" TTDD"→A$[1,5];gsb "SUBHEAD"
154: "/"→A$[10,10];gsb "ARRAY 15"
155: 0→W;1→D
156: gto +9
157: "BOUND":Y[1]→B[2]
158: if B[7]>B[1] and B[7]<B[2];B[7]→B[2]
159: if B[8]>B[1] and B[8]<B[2];B[8]→B[2]
160: if B[9]>B[1] and B[9]<B[2];B[9]→B[2]
161: if B[10]>B[1] and B[10]<B[2];B[10]→B[2]
162: if B[12]>B[1] and B[12]<B[2];B[12]→B[2]
163: if F[B[2],4]>=100;B[2]→B[1];gto -6
164: ret
165: 0→B[1]→F;gsb "BOUND"
166: 0→B[3]→A→B[4]
167: for M=B[1]+1 to B[2]
168: (F[M,6]-F[B[1],6])/(F[B[2],6]-F[B[1],6])→C
169: F[M,1]-F[B[1],1]-C(F[B[2],1]-F[B[1],1])→A
170: if prnd(F[M,1],-1)=prnd(B[7],-1);M→B[2];gto +4
171: if prnd(F[M,1],-1)=prnd(B[8],-1);M→B[2];gto +3
172: if abs(A)>2 and abs(A)>B[4];M→B[3];abs(A)→B[4]
173: next M
174: if B[3]>B[1] and B[3]<B[2];B[3]→B[2];gto -8
175: 0→B[3]→A;if B[1]>B[12];gto +8
176: for M=B[1]+1 to B[2]
177: (F[M,6]-F[B[1],6])/(F[B[2],6]-F[B[1],6])→C
178: F[M,2]-F[B[1],2]-C(F[B[2],2]-F[B[1],2])→A
179: if prnd(F[M,2],0)=prnd(B[9],0);M→B[2];gto +4
180: if prnd(F[M,2],0)=prnd(B[10],0);M→B[2];gto +3
181: if abs(A)>10 and F[M,2]>20;M→B[2];gto +2
182: next M
183: B[2]→F
184: if F[F,4]<1;gto +13
185: if F[F,4]>=100;gto +9
186: "//////////"→A$[1,10];if D>9;1→D
187: str(prnd(D11,0))→A$[1,3];A$[2,3]→A$[1,2];"/"→A$[3,3]
188: 10prnd(F[F,4],-1)→Q
189: gsb "STRING"
190: gsb "TEMP"
191: gsb "DP"
192: 1+D→D
193: gsb "ARRAY 10"
194: if F=Y[1];gto +3
195: F→B[1];gsb "BOUND"
196: gto -30
197: gsb "FOOT"
198: "035050070090100110140150160170"→V$[1,30]
199: spc 1;" PPDD"→A$[1,5];gsb "SUBHEAD"
*11136
```

```
200: "0"→A$[10,10];gsb "ARRAY 15"
201: int((E[Y[1],6]/10+val(P$[19,21]))*.03281)→r9
202: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
203: if int(M[1,N]*.003281)<r9;int(M[1,N]*.003281)→r9
204: int((val(Z$[6,8])+val(P$[19,21])+1000)*.03281)→r10
205: 1→D
206: for W=0 to 9
207: val(W$[W3+1,W3+3])→Q
208: if r10>Q;next W
209: if Q>r9;gto +12
210: Q*304.8→Q
211: gsb "ANGLES"
212: "9////"→I$[1,5];"                    "→F$[1,15]
213: if Q>=100*304.8;"1"→I$[1,1]
214: W$[W3+2,W3+3]→I$[2,3];A$[11,15]→F$[1,5]
215: str(r9)→B$;if r9<100 and B$[2,2]=W$[W3+2,W3+2];2→D;next W
216: if r9>=100 and B$[3,3]=W$[W3+2,W3+2];2→D;next W
217: I$[1,5]→A$[1,5];gsb "ARRAY 5"
218: F$[1,5]→A$[1,5];gsb "ARRAY 5"
219: next W
220: gto +10
221: r9*304.8→Q;gsb "ANGLES"
222: if r9=val(W$[W3-2,W3]);gto +8
223: str(r9)→B$;if r9<100;B$[2,3]→B$[3,4]
224: if D=1;"9////"→I$[1,5];"            "→F$[1,10]
225: if D=1 and Q>=100*304.8;"1"→I$[1,1]
226: if D=1;B$[3,4]→I$[2,3];A$[11,15]→F$[1,5]
227: if D=2;B$[4,4]→I$[4,4];A$[11,15]→F$[6,10]
228: I$[1,5]→A$[1,5];gsb "ARRAY 5"
229: F$[1,10]→A$[1,10];gsb "ARRAY 10"
230: gsb "FOOT"
231: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
232: trk 0;ldf 15
233: end
*19106
```

e. Last Flight.

(1) Sound Ranging, Computer Met, Fallout, Ballistic 3

TRACK 1 FILE 4

```
0: dsp "TURN-ON BEEPX";gsb "BBB"
1: dim A$[100],F$[5],R[10],C[0:16,1:14]
2: dim P$[102],O$[16],C$[16],L$[16],N$[6],C$[192],D$[16]
3: dim F[-1:60,1:6],Y[2]
4: dim B[-2:45,1:6],P[15]
5: dim H$[2,32];time 500
6: trk 1;ldf 5,P$,C$,O$,L$,N$,C$,D$
7: spc 5;prt "LAST FLIGHT:",D$[1,16]
8: sfg 4;if not ics2;cfg 4
9: if not ics2;gto +6
10: ent "NEED TTY-76 TELETAPE?",A
11: if A#-1tn^10 and A#-2tn^10;gto -1
12: if A=-1tn^10;"Y"→P$[101,101]
13: if A=-2tn^10;"N"→P$[101,101]
14: trk 1;rcf 5,P$,O$,C$,L$,N$,C$,D$
15: spc 5
16: dsp "LAST FLIGHT MESSAGE OUTPUT";gsb "BBB"
17: if P$[101,101]="Y";sfg 7;cfg 8
18: if P$[101,101]="N";cfg 7;sfg 8
19: trk 1;ldf 6,F[*],Y[*]
20: trk 1;ldf 7,H[*]
21: " 5 9 #,. )4&80:;3 $? 6!/-2´ 71( "→H$[1]
22: " T O HNM LRCIPCVFZDBSYFXAWJ UOK "→H$[2]
23: char(0)→H$[1,1,1];char(10)→H$[1,9,9];char(7)→H$[1,21,21]
24: char(13)→H$[1,3,3]
25: gsb "INTHT"
26: sfg 3;dsp "SOUND RANGING";gsb "BBB"
27: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
28: gto +69
29: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
30: "INTHT":for N=1 to Y[1];if F[N,3]#0;gto +7
31: for K=N to Y[1];if F[N,3]=0;next K
32: (F[K,5]-F[N,5])/(F[K,5]-F[N-1,5])→C
33: F[K,1]-C(F[K,1]-F[N-1,1])→F[N,1]
34: F[K,2]-C(F[K,2]-F[N-1,2])→F[N,2]
35: F[K,3]-C(F[K,3]-F[N-1,3])→F[N,3]
36: F[K,6]-C(F[K,6]-F[N-1,6])→F[N,6]
37: next N
38: ret
39: "FCOT":fmt 2,"9",/
40: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2;wtc 2,0
41: fmt b,z
42: if flg4 and flg7;par 0;wtc 2,2;wrt 2,3,2,8;wtc 2,0
43: if flg4;wtc 2,2;for N=1 to 30;wtb 2,0;next N;wtc 2,0
44: ret
45: "A→F":fmt b,z;par 0
46: for C=1 to B
47: for K=1 to 2;pos(H$[K],A$[C,C])→F;if F;sfg K;gto +2
48: next K
49: if F=1 or F=3 or F=5 or F=9;cfg 1,2,5,6;gto +5
*10734
```

```
50: if flg1 and flg5;cfg 1;gto +4
51: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
52: if flg2 and flg6;cfg 2;gto +2
53: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
54: wtc 2,2;wrt 2,P-1;wtc 2,0
55: next C
56: wtc 2,2;wrt 2,2,8;wtc 2,0
57: ret
58: "R→P":√(XX+YY)→R[1]
59: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
60: A→R[2]
61: if A<0;360+A→R[2]
62: ret
63: "READER":cfg 1,2,5,6;rar 2
64: if flg4 and flg8;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
65: if flg4 and flg7;12→B;gsb "A→B"
66: dsp A$[1,12];prt A$[1,12]
67: P$[85,99]→A$[1,J]
68: if flg9;cfg 9;gto +5
69: fmt 3,c12
70: if flg4 and flg8;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
71: if flg4 and flg7;12→B;gsb "A→B"
72: gto +3
73: if flg4 and flg8;fmt 3,c15;wtc 2,2;wrt 2.3,A$[1,15];wtc 2,0
74: if flg4 and flg7;15→B;gsb "A→B"
75: dsp A$[1,J];prt A$[1,J]
76: ret
77: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000
78: ret
79: "P→R":R[1]*cos(R[2])→X;R[1]*sin(R[2])→Y
80: ret
81: "STRING":fxd 0;str(M)→B$[1,5]
82: if M<1000;B$[2,4]→B$[3,5];"0"→B$[2,2]
83: if M<100;B$[2,4]→B$[3,5];"0"→B$[2,2]
84: if M<10;B$[2,4]→B$[3,5];"0"→B$[2,2]
85: if M<1;B$[2,4]→B$[3,5];"0"→B$[2,2]
86: ret
87: "ARRAY":I→M;gsb "STRING"
88: B$[4,5]→A$[1,2];A→M;gsb "STRING"
89: B$[3,5]→A$[3,5];S→M;gsb "STRING"
90: B$[3,5]→A$[6,8];T→M;gsb "STRING"
91: B$[2,5]→A$[9,12];C→M;gsb "STRING"
92: B$[2,5]→A$[13,16]
93: if flg4 and flg8;rar 2;fmt 4,c16;wtc 2,2;wrt 2.4,A$[1,16];wtc 2,0
94: if flg4 and flg7;16→B;gsb "A→B"
95: dsp A$[1,16];prt A$[1,16]
96: ret
97: B[-1,5]→R[2];B[-1,6]→R[1];gsb "P→R"
98: X→R[4];Y→R[6]
99: rnd(B[-2,5]*16/9,0)→A→R[10]
*30887
```

```
100: prnd(P[-2,6],0)→S→P[9]
101: prnd(B[-1,4]10,0)→T
102: .000001→E[0,4]
103: 0→I
104: if E[5,4]=0;sto 5;prt "SOUND EMCINC   ECS DEAFY";gto +48
105: for P=1 to 5;if E[P,4]=0;gto +13
106: P-1→J
107: if P=3;next P
108: if P=4;2→J
109: 1+I→I
110: 6367650→S
111: √((S+B[P,1])^2-S^2cos(P[P,6])^2)-Ssin(B[P,6])→E[8]
112: (Scos(B[P,6])/(S+B[P,1]))K[8]→K[I]
113: B[P,5]→E[2];gsb "E→K"
114: E[4]→P[5];X→E[4];P[5]-E[4]→X;E[6]→E[7];Y→E[6];E[7]-E[6]→Y;gsb "E→P"
115: prnd((P[I]/(B[P,4]-B[J,4])).03238,0)→S→P[I]
116: prnd(P[2]*16/9,0)→A→E[I+4]
117: next P
118: if P[2]<P[1] and (P[2]>=P[9]+2 or P[2]<=P[9]-2);P[2]→C;P[6]→A;gto +16
119: if P[2]<P[1] and (P[2]<P[9]+2 and P[2]>P[9]-2);P[1]→C;P[5]→A;gto +15
120: if P[2]>2*P[1];.4→r1;0→r2;.3→r3;.15→r4;.15→r5;gto +2
121: .2→r1;.5→r2;.35→r3;.075→r4;.075→r5
122: 0→X→Y→C→D
123: r1P[9]→P[1];P[10]*9/16→P[2];gsb "P→R"
124: X→C;Y→D
125: r2P[1]→P[1];P[5]*9/16→P[2];gsb "P→R"
126: X+C→C;Y+D→D
127: r3P[2]→P[1];P[6]*9/16→P[2];gsb "P→R"
128: X+C→C;Y+D→D
129: r4P[3]→P[1];P[7]*9/16→P[2];gsb "P→R"
130: X+C→C;Y+D→D
131: r5P[4]→P[1];P[8]*9/16→P[2];gsb "P→R"
132: X+C→X;Y+D→Y;gsb "R→P"
133: P[1]→S;P[2]*16/9→A
134: prnd(S,0)→S;prnd(A,0)→A
135: if A=0;64→A
136: if S=0;0→A
137: for L=1 to 5;if not (P[L,6]>200 and P[L-1,6]<200);next L
138: (200-P[L-1,6])/(P[L,6]-P[L-1,6])→G
139: G(P[L,3]-P[L-1,3])+P[L-1,3]→B[16,3]
140: G(P[L,1]-P[L-1,1])+P[L-1,1]+273.16+3B[16,3]→B[16,3]
141: B[16,3]/4-273.16→B[16,3]→T
142: prnd(T*10,0)→T
143: abs(T)→M;" "→A$[1,1];if T<0;"-"→A$[1,1]
144: gsb "STRING"
145: B$[3,5]→A$[2,4];A→M;gsb "STRING"
146: B$[3,5]→A$[5,7];S→M;gsb "STRING"
147: B$[4,5]→A$[8,9]
148: if not flg3;gto +5
149: A$[1,9]→B$[91,99];B$[73,84]→A$[1,12];15→J
*5506
```

```
150: sfg 9;gsb "HEADER"
151: cfg 9;gsb "FOOT"
152: spc 5
153: sfg 3;dsp "COMPUTER NET";gsb "RRR"
154: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→R"
155: X→R[4];Y→R[6]
156: str(prnd(H[-1,2],0))→A$;A$[2,4]→P$[22,24]
157: if H[-1,2]>=1000;A$[3,5]→P$[22,24]
158: P$[1,12]→A$[1,12];P$[13,24]→B$[85,96];12→J
159: if not flg3;gto +34
160: gsb "HEADER"
161: 0→I
162: prnd(H[-2,5]*16/9,0)→A;if A=0;64→A
163: prnd(H[-2,6],0)→S;if S=0;0→A
164: prnd(H[-1,4]10,0)→T
165: prnd(H[-1,2],0)→Q
166: gsb "ARRAY"
167: .0000001→H[0,4]
168: for P=1 to 29;if H[P,4]=0;gto +23
169: if P=1;0→J
170: if P=2;next P
171: if P=3;1→J
172: if P=4 or P=5;next P
173: if P=6;P-3→J
174: if P>6;P-1→J
175: 1+I→I
176: 6367650→S
177: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
178: (Scos(H[P,6])/(S+H[P,1]))R[8]→R[1]
179: H[P,5]→R[2];gsb "P→R"
180: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "P→P"
181: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S
182: H[0,3]H[P,1]6371299/(6371299+H[P,1])→B[1]
183: H[0,3]H[J,1]6371299/(6371299+H[J,1])→B[2]
184: prnd(((B[1]-B[2])/67.442(H[J,2]-H[P,2]))10,0)→T
185: tr^((H[P,2]+H[J,2])/2)→Q;prnd(Q,0)→Q
186: prnd(R[2]*16/9,0)→A
187: if A=0;64→A
188: if S=0;0→A
189: gsb "ARRAY"
190: next P
191: gsb "FOOT"
192: spc 5
193: sfg 3;dsp "FALLOUT NET";gsb "RRR"
194: str(prnd(1000H[-1,2]/1013.25,0))→A$;A$[2,4]→P$[22,24]
195: if H[-1,2]>=1013.25;A$[3,5]→P$[22,24]
196: gto +11
197: "ARRAY-12":I→M;gsb "STRING"
198: B$[4,5]→A$[1,2];A→M;gsb "STRING"
199: B$[4,5]→A$[3,4];S→M;gsb "STRING"
*20897
```

```
200: B$[4,5]→A$[5,6];T→N;gsb "STRING"
201: B$[3,5]→A$[7,9];Q→N;gsb "STRING"
202: B$[3,5]→A$[10,12]
203: if flg4 and flg8;par 2;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
204: if flg4 and flg7;12→B;gsb "A→B"
205: dsp A$[1,12];prt A$[1,12]
206: ret
207: H[-1,5]→P[2];H[-1,6]→P[1];gsb "P→P"
208: X→P[4];Y→P[6]
209: 1→I
210: H[-1,2]→C[0,4]
211: for P=1 to 27;if H[P,4]=0;gto +23
212: P-2→J
213: if P=1;0→J
214: if P=2;next P
215: if P=3;1→J
216: if P=4 or P=5;next P
217: if P=6;P-3→J
218: if P=7 or P=8 or P=15;P-1→J
219: if P=9 or P=11 or P=13 or P=16 or P=18;next P
220: if P=20 or P=22 or P=24 or P=26;next P
221: 6371299→S
222: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→P[8]
223: (Scos(H[P,6])/(S+H[P,1]))P[8]→P[1]
224: H[P,5]→P[2];gsb "P→P"
225: P[4]→P[5];X→P[4];P[5]-P[4]→X;P[6]→P[7];Y→P[6];P[7]-P[6]→Y;gsb "P→P"
226: prnd((P[1]/(H[P,4]-H[J,4])).03238,0)→S→C[I,2]
227: H[0,3]P[P,1]6371299/(6371299+H[P,1])→P[1]
228: H[0,3]H[J,1]6371299/(6371299+H[J,1])→P[2]
229: (P[1]-P[2])/67.442(H[J,2]-H[P,2])→C[I,3];prnd(C[I,3],0)→T
230: tn^((H[P,2]+H[J,2])/2)→Q→C[I,4];prnd(Q,0)→Q
231: P[2]→C[I,5]
232: if I<=14;I+1→I→C[I,1]
233: next P
234: for Z=1 to 16;if C[Z,4]#0;next Z
235: Z-1→Z
236: H[-1,4]→C[0,3]
237: for N=0 to Z;348.38395C[N,4]/C[N,3]→C[N,9];next N
238: C[0,9]/1225→C[0,10];C[1,9]/1213.3→C[1,10];C[2,9]/1184.4→C[2,10]
239: C[3,9]/1139.2→C[3,10];C[4,9]/1084.6→C[4,10];C[5,9]/1032→C[5,10]
240: C[6,9]/957→C[6,10];C[7,9]/863.4→C[7,10];C[8,9]/777→C[8,10]
241: C[9,9]/697.4→C[9,10];C[10,9]/590→C[10,10];C[11,9]/467→C[11,10]
242: C[12,9]/364.8→C[12,10];C[13,9]/266.6→C[13,10];C[14,9]/194.8→C[14,10]
243: C[15,9]/142.3→C[15,10]
244: 100C[0,10]→C[0,11]
245: H[-1,4]→C[0,3]
246: C[0,3]/288.2→C[0,6]
247: C[1,3]/287.5→C[1,6];C[2,3]/285.9→C[2,6];C[3,3]/283.3→C[3,6]
248: C[4,3]/280→C[4,6];C[5,3]/276.8→C[5,6];C[6,3]/271.9→C[6,6]
249: C[7,3]/265.5→C[7,6];C[8,3]/259→C[8,6];C[9,3]/252.5→C[9,6]
*5846
```

```
250: C[10,3]/242.7→C[10,6];C[11,3]/229.8→C[11,6];C[12,3]/216.8→C[12,6]
251: C[13,3]/216.7→C[13,6];C[14,3]/216.7→C[14,6];C[15,3]/216.7→C[15,6]
252: gto +9
253: "ARRAY-FO":I→M;gsb "STRINC"
254: B$[4,5]→A$[1,2];A→N;gsb "STRINC"
255: B$[3,5]→A$[3,5];S→M;gsb "STRINC"
256: B$[3,5]→A$[6,8]
257: if flg4 and flg8;par 2;fmt 1,c8;wtc 2,2;wrt 2.1,A$[1,8];wtc 2,0
258: if flg4 and flg7;8→B;gsb "A→E"
259: dsp A$[1,8];prt A$[1,8]
260: ret
261: H[-1,5]→R[2];H[-1,6]→R[1];gsb "P→R"
262: X→R[4];Y→R[6]
263: prnd(H[-2,5]*16/9,0)→A;if A=0;64→A
264: prnd(H[-2,6],0)→S;if S=0;0→A
265: P$[49,60]→A$[1,12];P$[61,72]→P$[85,96];12→J
266: if not flg3;gto +28
267: gsb "HEADER"
268: 0→I
269: gsb "ARRAY-FO"
270: 1→I
271: for P=8 to 34;if H[P,4]=0;gto +21
272: if P=8;0→J
273: if P=9 or P=10 or P=11;next P
274: if P=12;8→J
275: if P=13 or P=14;next P
276: if P=15;12→J
277: if P=16 or P=18 or P=20 or P=22 or P=24 or P=26 or P=28;next P
278: if P=17 or P=19 or P=21 or P=23 or P=25 or P=27 or P=29;P-2→J
279: if P>29;P-1→J
280: 6371299→S
281: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→R[8]
282: (Scos(H[P,6])/(S+H[P,1]))R[8]→R[1]
283: H[P,5]→R[2];gsb "P→R"
284: R[4]→R[5];X→R[4];R[5]-R[4]→X;R[6]→R[7];Y→R[6];R[7]-R[6]→Y;gsb "P→R"
285: prnd((R[1]/(H[P,4]-H[J,4])).03238,0)→S
286: prnd(R[2]*16/9,0)→A
287: if A=0;64→A
288: if S=0;0→A
289: gsb "ARRAY-FO"
290: 1+I→I
291: next P
292: gsb "FOOT"
293: spc 5
294: sfg 3;dsp "BALLISTIC TYPE 3";gsb "PBP"
295: gto +11
296: "T3":
297: B+1→F;for N=1 to Z;rNC[F,6]+C[N,7]→C[N,7];next N
298: ret
299: "D3":
*11567
```

```
300: E+1→E;for N=1 to Z;rNC[E,10]+C[N,11]→C[N,11];next N
301: ret
302: "WW":
303: C[1,2]→P[1];C[I,5]→P[2];I+1→I;gsb "D→E"
304: for N=1 to Z;XrN+C[N,13]→C[N,13];YrN+C[N,14]→C[N,14];next N
305: ret
306: 100C[0,6]→C[0,7]
307: 100→r1;27→r2;13→r3;8→r4;5→r5;4→r6;2→r7;1→r8→r9→r10→r11→r12→r13→r14→r15
308: gsb "T3"
309: 0→r1;73→r2;20→r3;12→r4;10→r5;4→r6→r7;3→r8;gsb "T3"
310: 0→r2;67→r3;25→r4;20→r5;9→r6;7→r7;5→r8;2→r9→r10→r11→r12→r13→r14→r15
311: gsb "T3"
312: 0→r3;55→r4;21→r5;11→r6;9→r7;4→r8;3→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
313: 0→r4;44→r5;13→r6;11→r7;10→r8;gsb "T3"
314: 0→r5;59→r6;26→r7;19→r8;9→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
315: 0→r6;41→r7;23→r8;13→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
316: 0→r7;35→r8;24→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
317: 0→r8;44→r9→r10→r11→r12→r13→r14→r15;gsb "T3"
318: 100→r1;43→r2;22→r3;15→r4;11→r5;8→r6;6→r7;5→r8;4→r9;3→r10;1→r11
319: 2→r12→r13→r14→r15;0→E;gsb "D3"
320: 0→r1;57→r2;31→r3;21→r4;17→r5;11→r6;8→r7;6→r8;6→r9;4→r10;3→r11→r12→r14
321: 4→r15;gsb "D3"
322: 0→r2;47→r3;32→r4;25→r5;17→r6;14→r7;11→r8;9→r9;7→r10;5→r11→r12→r14→r15
323: 4→r13;gsb "D3"
324: 0→r3;32→r4;22→r5;17→r6;13→r7;5→r13;gsb "D3"
325: 0→r4;25→r5;15→r6;12→r7;10→r8;8→r9;6→r11;gsb "D3"
326: 0→r5;32→r6;22→r7;19→r8;17→r9;13→r10;12→r11;11→r12→r13;10→r14→r15;gsb "D3"
327: 0→r6;25→r7;17→r8;15→r9;12→r10;11→r11;10→r12;9→r13→r14→r15;gsb "D3"
328: 0→r7;21→r8;14→r9;11→r10;9→r11→r12;8→r14→r15;gsb "D3"
329: 0→r8;18→r9;11→r10;9→r11;8→r12→r13;7→r14→r15;gsb "D3"
330: 0→r9;25→r10;16→r11;14→r12→r13;13→r14;12→r15;gsb "D3"
331: 0→r10;23→r11;12→r12;10→r13;11→r14;9→r15;gsb "D3"
332: 0→r11;16→r12;9→r13;8→r14→r15;gsb "D3"
333: 0→r12;12→r13;6→r14;5→r15;gsb "D3"
334: 0→r13;8→r14;5→r15;gsb "D3"
335: 0→r14;6→r15;gsb "D3"
336: 1→I
337: 100→r1;20→r2;9→r3;6→r4;4→r5;3→r6;2→r7→r8→r9;1→r10;0→r11→r12→r15;gsb "WW"
338: 0→r1;80→r2;19→r3;12→r4;8→r5;5→r6;3→r7;2→r10;1→r12→r13→r14→r15;gsb "WW"
339: 0→r2;72→r3;26→r4;15→r5;8→r6;7→r7;6→r8;5→r9;1→r11;gsb "WW"
340: 0→r3;56→r4;20→r5;9→r6;4→r10→r11;2→r12;gsb "WW"
341: 0→r4;53→r5;12→r6;8→r7;3→r10→r11;4→r12;3→r13;2→r14→r15;gsb "WW"
342: 0→r5;63→r6;20→r7;14→r8;12→r9;7→r10→r12→r13→r14→r15;8→r11;gsb "WW"
343: 0→r6;53→r7;19→r8;13→r9;8→r10→r11;7→r12;gsb "WW"
344: 0→r7;45→r8;20→r9;9→r10→r11;gsb "WW"
345: 0→r8;36→r9;8→r12;gsb "WW"
346: 0→r9;55→r10;20→r11;17→r12;15→r13;13→r14;12→r15;gsb "WW"
347: 0→r10;38→r11;16→r12;14→r13;gsb "WW"
348: 0→r11;30→r12;13→r13;11→r15;gsb "WW"
349: 0→r12;24→r13;10→r14→r15;gsb "WW"
*26295
```

```
350: 0→r13;18→r14;8→r15;gsb "WW"
351: 0→r14;14→r15;gsb "WW"
352: for N=1 to Z;C[N,13]→X;C[N,14]→Y;gsb "F→P"
353: F[1]→C[N,13];F[2]→C[N,14];next N
354: H[-2,5]→C[0,14];H[-2,6]100→C[0,13]
355: if not flg3;gto +14
356: P$[25,36]→A$[1,12];P$[13,24]→P$[85,96];12→J;gsb "HLADEP"
357: for N=0 to Z;N→I
358: rrnd(C[N,11]10,0)→C
359: rrnd(C[N,7]10,0)→T
360: rrnd(C[N,13]/100,0)→S
361: rrnd(C[N,14]*16/90,0)→A
362: if A=0;64→A
363: if S=0;0→A
364: if S>99;S-100→S;I+80→I
365: gsb "ARRAY-12"
366: next N
367: gsb "ROOT"
368: src 5
369: trk 1;ldf 8
370: end
*17276
```

(2) Ballistic 2

TRACK 1 FILE 8

```
0: sfg 3;dsp "BALLISTIC TYPE 2";gsb "BBB"
1: gto +79
2: "BEEP":beep;wait 150;beep;wait 150;beep;wait 3000;ret
3: "BOOT":fmt 2,"9",/
4: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2;wtc 2,0
5: fmt b,z
6: if flg4 and flg7;par 0;wtc 2,2;wrt 2,3,2,8;wtc 2,0
7: if flg4;wtc 2,2;for N=1 to 30;wtb 2,0;next N;wtc 2,0
8: ret
9: "A→B":fmt b,z;par 0
10: for C=1 to B
11: for K=1 to 2;pos(B$[K],A$[C,C])→B;if B;sfg K;gto +2
12: next K
13: if B=1 or B=3 or B=5 or B=9;cfg 1,2,5,6;gto +5
14: if flg1 and flg5;cfg 1;gto +4
15: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
16: if flg2 and flg6;cfg 2;gto +2
17: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
18: wtc 2,2;wrt 2,B-1;wtc 2,0
19: next C
20: wtc 2,2;wrt 2,2,8;wtc 2,0
21: ret
22: "P→R":√(XX+YY)→R[1]
23: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
24: A→R[2]
25: if A<0;360+A→R[2]
26: ret
27: "READER":cfg 1,2,5,6;par 2
28: if flg4 and flg8;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
29: if flg4 and flg7;12→B;gsb "A→B"
30: dsp A$[1,12];prt A$[1,12]
31: P$[85,99]→A$[1,J]
32: if flg9;gto +4
33: if flg4 and flg8;wtc 2,2;wrt 2.3,A$[1,J];wtc 2,0
34: if flg4 and flg7;12→B;gsb "A→B"
35: gto +3
36: if flg4 and flg8;fmt 3,c15;wtc 2,2;wrt 2.3,A$[1,J];wtc 2,0
37: if flg4 and flg7;15→B;gsb "A→B"
38: dsp A$[1,J];prt A$[1,J]
39: ret
40: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000
41: ret
42: "P→R":R[1]*cos(R[2])→X;R[1]*sin(R[2])→Y
43: ret
44: "STRING":fxd 0;str(M)→B$[1,5]
45: if M<1000;B$[2,4]→B$[3,5];"0"→B$[2,2]
46: if M<100;B$[2,4]→B$[3,5];"0"→B$[2,2]
47: if M<10;B$[2,4]→B$[3,5];"0"→B$[2,2]
48: if M<1;B$[2,4]→B$[3,5];"0"→B$[2,2]
49: ret
*23874
```

```
50: "ARRAY":I→N;gsb "STRINC"
51: B$[4,5]→A$[1,2];A→N;gsb "STRINC"
52: B$[3,5]→A$[3,5];S→N;gsb "STRINC"
53: B$[3,5]→A$[6,8];T→N;gsb "STRINC"
54: B$[2,5]→A$[9,12];C→N;gsb "STRINC"
55: B$[2,5]→A$[13,16]
56: if flg4 and flg8;par 2;fmt 4,c16;wtc 2,2;wrt 2.4,A$[1,16];wtc 2,0
57: if flg4 and flg7;16→R;gsb "A→B"
58: dsp A$[1,16];prt A$[1,16]
59: ret
60: "ARRAY-12":I→N;gsb "STRINC"
61: B$[4,5]→A$[1,2];A→N;gsb "STRINC"
62: B$[4,5]→A$[3,4];S→N;gsb "STRINC"
63: B$[4,5]→A$[5,6];T→N;gsb "STRINC"
64: B$[3,5]→A$[7,9];C→N;gsb "STRINC"
65: B$[3,5]→A$[10,12]
66: if flg4 and flg8;par 2;fmt 3,c12;wtc 2,2;wrt 2.3,A$[1,12];wtc 2,0
67: if flg4 and flg7;12→R;gsb "A→B"
68: dsp A$[1,12];prt A$[1,12]
69: ret
70: "T2":
71: E+1→E;for N=1 to Z;rNC[E,6]+C[N,8]→C[N,8];next N
72: ret
73: "D2":
74: E+1→E;for N=1 to Z;rNC[E,10]+C[N,12]→C[N,12];next N
75: ret
76: "VV":
77: C[I,2]→F[1];C[I,5]→F[2];1+I→I;gsb "F→F"
78: for N=1 to Z;XrN+C[N,13]→C[N,13];YrN+C[N,14]→C[N,14];next N
79: ret
80: 100C[0,6]→C[0,8]
81: 100→r1;63→r2;37→r3;25→r4;20→r5;13→r6;10→r7;9→r8;7→r9;5→r10→r11→r13→r14
82: 4→r12;5→r15;0→E;gsb "T2"
83: 0→r1;37→r2;37→r3;30→r4;24→r5;19→r6;14→r7;10→r8;9→r9;8→r10
84: 6→r11→r12→r13→r14→r15;gsb "T2"
85: 0→r2;26→r3;35→r4;30→r5;24→r6;20→r7;17→r8;14→r9;12→r10;10→r11→r12→r13→r14
86: 10→r15;gsb "T2"
87: 0→r3;10→r4;18→r5→r6;16→r7;15→r8;13→r9;10→r10;8→r12
88: 9→r11→r13→r14→r15;gsb "T2"
89: 0→r4;8→r5;14→r6→r7;13→r8;12→r9;10→r10;8→r11→r13→r14→r15;gsb "T2"
90: 0→r5;12→r6;19→r7;20→r8;19→r9;17→r10;15→r11;14→r12;16→r13→r14→r15;gsb "T2"
91: 0→r6;7→r7;12→r8;15→r9;14→r10;13→r11→r12;12→r13→r14→r15;gsb "T2"
92: 0→r7;4→r8;8→r9;10→r10;12→r11;11→r12;13→r13→r14→r15;gsb "T2"
93: 0→r8;3→r9;8→r10;10→r11→r12;11→r13→r14→r15;gsb "T2"
94: 0→r9;6→r10;12→r11;16→r12;10→r13→r14→r15;gsb "T2"
95: 100C[0,10]→C[0,12]
96: 100→r1;63→r2;37→r3;25→r4;20→r5;13→r6;10→r7;9→r8;7→r9;5→r10;4→r11→r12
97: 3→r13→r14;2→r15;0→E;gsb "D2"
98: 0→r1;37→r2;37→r3;30→r4;24→r5;19→r6;14→r7;10→r8;9→r9;8→r10;6→r11→r12
99: 5→r13→r14→r15;gsb "D2"
*32459
```

```
100: 0→r2;26→r3;35→r4;30→r5;24→r6;20→r7;17→r8;14→r9;12→r10;10→r11;9→r12
101: 8→r13;6→r14→r15;gsb "D2"
102: 0→r3;10→r4;18→r5→r6;16→r7;15→r8;13→r9;10→r10;8→r11→r12
103: 7→r14→r15;gsb "D2"
104: 0→r4;8→r5;14→r6→r7;13→r8;12→r9;6→r13;5→r15;gsb "D2"
105: 0→r5;12→r6;19→r7;20→r8;19→r9;17→r10;15→r11;13→r12;12→r13;11→r14→r15
106: gsb "D2"
107: 0→r6;7→r7;12→r8;15→r9;14→r10;13→r11;12→r12;11→r13;10→r14→r15;gsb "D2"
108: 0→r7;4→r8;8→r9;10→r10→r11→r12→r13;9→r14;8→r15;gsb "D2"
109: 0→r8;3→r9;8→r10;10→r11;8→r12→r13→r14→r15;gsb "D2"
110: 0→r9;6→r10;12→r11;13→r12→r13→r14→r15;gsb "D2"
111: 0→r10;4→r11;7→r12;9→r13;10→r14→r15;gsb "D2"
112: 0→r11;2→r12;5→r13;6→r14;7→r15;gsb "D2"
113: 0→r12;2→r13;4→r14;5→r15;gsb "D2"
114: 0→r13;1→r14;3→r15;gsb "D2"
115: for N=1 to Z;0→C[N,13]→C[N,14];next N
116: 1→I
117: 100→r1;50→r2;29→r3;18→r4;13→r5;8→r6;7→r7;4→r8→r9;3→r10→r12;1→r15
118: 2→r11→r13→r14;gsb "WW"
119: 0→r1;50→r2;33→r3;23→r4;18→r5;12→r6;8→r7→r8;6→r9;3→r15
120: 4→r10→r11→r12→r13→r14;gsb "WW"
121: 0→r2;38→r3;39→r4;31→r5;22→r6;16→r7;13→r8;11→r9;8→r10;6→r11;7→r12
122: 5→r13→r14→r15;gsb "WW"
123: 0→r3;20→r4;27→r5;20→r6;15→r7;12→r8;10→r9;7→r11;6→r13→r14;4→r15;gsb "WW"
124: 0→r4;11→r5;19→r6;16→r7;13→r8;6→r11;4→r14;5→r15;gsb "WW"
125: 0→r5;19→r6;27→r7;24→r8;21→r9;16→r10;13→r11;12→r12;11→r13;9→r14→r15
126: gsb "WW"
127: 0→r6;11→r7;18→r8;20→r9;15→r10;11→r12;10→r13;gsb "WW"
128: 0→r7;8→r8;12→r9;14→r10;12→r11;10→r12;9→r13;8→r15;gsb "WW"
129: 0→r8;6→r9;13→r10;11→r11;8→r12→r13→r14;7→r15;gsb "WW"
130: 0→r9;11→r10;18→r11;15→r12;14→r13;13→r14;12→r15;gsb "WW"
131: 0→r10;8→r11;10→r12;11→r13→r14;10→r15;gsb "WW"
132: 0→r11;6→r12;9→r13→r14→r15;gsb "WW"
133: 0→r12;5→r13;6→r14;8→r15;gsb "WW"
134: 0→r13;5→r14;6→r15;gsb "WW"
135: 0→r14;4→r15;gsb "WW"
136: for N=1 to Z;C[N,13]→X;C[N,14]→Y;gsb "P→P"
137: P[1]→C[N,13];P[2]→C[N,14];next N
138: P[-2,5]→C[0,14];P[-2,6]100→C[0,13]
139: if not flg2;gto +14
140: PS[37,48]→AS[1,12];PS[13,24]→PS[85,96];12→J;gsb "HEADER"
141: for N=0 to Z;N→I
142: prnd(C[N,12]10,0)→C
143: prnd(C[N,8]10,0)→T
144: prnd(C[N,13]/100,0)→S
145: prnd(C[N,14]*16/90,0)→A
146: if A=0;64→A
147: if S=0;0→A
148: if S>99;S-100→S;I+80→I
149: gsb "ARRAY-12"
*10299
```

```
150: next N
151: gsb "FOOT"
152: spc 5
153: if P$[100,100]="N";trk 0;ldp 0,0,6
154: dim T[4];"////////"→C$[6,13]
155: 6.11*10^(7.5E[0,1]/(237.3+E[0,1]))→V
156: if E[0,4]<850;gto +3
157: if E[0,4]<1000 and E[0,4]>=850;"N"→C$[6,6];gto +12
158: if E[0,4]>=1000;"N///N"→C$[6,10];gto +16
159: E[0,1]+.07(850-E[0,4])+273.16→T[2]
160: T[2](1+.151212V/850)→T[4]
161: 6.11*10^(7.5(T[2]-273.16)/(237.3+T[2]-273.16))→V
162: T[2]+(.07)150→T[1]
163: T[1](1+.151212V/1000)→T[3]
164: 10val(P$[19,21])-33.721(T[4]+E[0,3])*log(850/E[0,4])→M;gsb "STPINC"
165: P$[3,5]→C$[6,8]
166: M-33.721(T[3]+T[4])*log(1000/850)→M;gsb "STPINC"
167: if M<0;M+500→M;str(M)→C$[10,13];C$[11,13]→C$[10,12];gto +8
168: if M>0;gsb "STPINC";P$[3,5]→C$[10,12];gto +6
169: E[0,1]+.07(1000-E[0,4])+273.16→T[1]
170: T[1](1+.151212V/1000)→T[3]
171: 10val(P$[19,21])-33.721(T[3]+E[0,3])*log(1000/E[0,4])→M;gsb "STPINC"
172: if M<0;M+500→M;str(M)→C$[10,13];C$[11,13]→C$[10,12];gto +2
173: P$[3,5]→C$[10,12]
174: trk 1;rcf 5,P$,C$,O$,L$,N$,C$,D$
175: trk 1;ldp 10
176: end
*19598
```

(3) AWS Output First Transmission

TRACK 1 FILE 10

```
0: dim A$[20],B$[15],C[1],E[12]
1: dim P$[102],O$[16],O$[16],L$[16],N$[6],C$[192],D$[16]
2: dim H[-2:45,1:6]
3: dim F[-1:60,1:6],Y[2]
4: dim M[3,45],C$[16]
5: dim V$[40],Z$[16],A[6]
6: dim I$[5],P$[10]
7: dim H$[2,32];time 500
8: cfg 4;if ios2;sfg 4
9: sfg 3;dsp "USAF AWS MESSAGE";gsb "BBB"
10: trk 1;ldf 5,B$,O$,O$,L$,N$,C$,D$
11: trk 1;ldf 6,F[*],Y[*]
12: trk 1;ldf 7,H[*]
13: if P$[101,101]="Y";sfg 7;cfg 8
14: if P$[101,101]="N";cfg 7;sfg 8
15: " 5 9 #,. )4&80:;3 $? 6!/-2' 71( "→H$[1]
16: " T O LNM LRCIPCVPZDESYPXAWJ UQK "→H$[2]
17: char(0)→H$[1,1,1];char(10)→H$[1,9,9];char(7)→H$[1,21,21]
18: char(13)→H$[1,3,3]
19: gsb "INTHT"
20: gto +125
21: "BBB":beep;wait 150;beep;wait 150;beep;wait 3000;ret
22: "INTHT":for N=1 to Y[1];if F[N,3]#0;gto +7
23: for K=N to Y[1];if F[N,3]=0;next K
24: (F[K,5]-F[N,5])/(F[K,5]-F[N-1,5])→C
25: F[K,1]-C(F[K,1]-F[N-1,1])→F[N,1]
26: F[K,2]-C(F[K,2]-F[N-1,2])→F[N,2]
27: F[K,3]-C(F[K,3]-F[N-1,3])→F[N,3]
28: F[K,6]-C(F[K,6]-F[N-1,6])→F[N,6]
29: next N
30: ret
31: "A→P":fmt b,z;par 0
32: for O=1 to r14
33: for K=1 to 2;pos(H$[K],C$[O,O])→J;if J;sfg K;gto +2
34: next K
35: if J=1 or J=3 or J=5 or J=9;cfg 1,2,5,6;gto +5
36: if flg1 and flg5;cfg 1;gto +4
37: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
38: if flg2 and flg6;cfg 2;gto +2
39: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
40: wtc 2,2;wrt 2,J-1;wtc 2,0
41: next O
42: ret
43: "FOOD":if flg4 and flg7;par 0;fmt 2,b,z;wtc 2,2;wrt 2.2,15;wtc 2,0
44: fmt 2,z,cl
45: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2,char(59);wtc 2,0
46: dsp ";";prt ";"
47: 10→C[1]
48: ret
49: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000;ret
*5131
```

```
50: "ARRAY 15":
51: "        "→C$[1,6]
52: A$[1,5]→C$[2,6];gsb "ARRAY 6"
53: A$[6,10]→C$[2,6];gsb "ARRAY 6"
54: A$[11,15]→C$[2,6];gsb "ARRAY 6"
55: dsp A$[1,15];prt A$[1,15]
56: ret
57: "ARRAY 10":
58: "        "→C$[1,6]
59: A$[1,5]→C$[2,6];gsb "ARRAY 6"
60: A$[6,10]→C$[2,6];gsb "ARRAY 6"
61: dsp A$[1,10];prt A$[1,10]
62: ret
63: "ARRAY 5":
64: "        "→C$[1,6]
65: A$[1,5]→C$[2,6];gsb "ARRAY 6"
66: dsp A$[1,5];prt A$[1,5]
67: ret
68: "ARRAY 6":
69: if C$[1,6]="        ";ret
70: C[1]+1→C[1];if C[1]<10;gto +7
71: if flg4 and flg8;rar 2;fmt 4,z,b;wtc 2,2;wrt 2.4,14],10;wtc 2,0
72: if flg4 and flg7;fmt 4,b,z;rar 0;wtc 2,2;wrt 2.4,2,8;wtc 2,0
73: 0→C[1];C$[2,6]→C$[1,5]
74: if flg4 and flg7;5→rl4;gsb "A→B"
75: if flg4 and flg8;rar 2;fmt 4,z,c5;wtc 2,2;wrt 2.4,C$[1,5];wtc 2,0
76: gto +3
77: if flg4 and flg8;rar 2;fmt 4,z,c6;wtc 2,2;wrt 2.4,C$[1,6];wtc 2,0
78: if flg4 and flg7;6→rl4;gsb "A→B"
79: "        "→C$[1,6]
80: ret
81: "STRING":if C=0;"000"→A$[3,5]
82: if C>999;str(C)→A$[6,10];A$[8,10]→A$[3,5]
83: if C>99 and C<1000;str(C)→A$[6,9];A$[7,9]→A$[3,5]
84: if C>9 and C<100;str(C)→A$[3,5];"0"→A$[3,3]
85: if C<10;str(C)→A$[4,5];"00"→A$[3,4]
86: ret
87: "TEMP":rnd(B[B,1],-1)→T;frc(T)→C
88: if sgn(T)=1 or sgn(T)=0;if C=.1 or C=.3 or C=.5 or C=.7 or C=.9;T-.1→T
89: if sgn(T)=-1;if C=.2 or C=.4 or C=.6 or C=.8 or C=0;T-.1→T
90: abs(T)→T
91: "000"→A$[6,9];str(10T)→C$
92: if T>0 and T<1;C$[2,2]→A$[8,8]
93: if T>.9 and T<10;C$[2,3]→A$[7,8]
94: if T>9.9;C$[2,4]→A$[6,8]
95: ret
96: "DP":if rnd(B[B,1],-1)<-40;"//"→A$[9,10];ret
97: if B[B,2]=0;2→B[B,2]
98: (B[B,2]/100)6.11tn^(7.5B[B,1]/(237.3+B[B,1]))→B
99: (237.31ce(B)-186.527)/(8.286-lce(B))→B
*8238
```

```
100: rrnd(F[F,1]-F,-1)→F
101: "00"→A$[9,10]
102: if F>0 and F<1;str(rrnd(10F,0))→C$;C$[2,2]→A$[10,10]
103: if F>.9 and F<5;str(rrnd(10F,0))→C$;C$[2,3]→A$[9,10]
104: if F>4.9;str(rrnd(50+F,0))→C$;C$[2,3]→A$[9,10]
105: ret
106: "ANGLES":for N=2 to 44;if Q>M[1,N];next N
107: if M[3,N]=0 and M[2,N]=0;"      "→A$[11,15];ret
108: M[1,N-1]→r1;M[1,N]→r2;M[3,N-1]→r3;M[3,N]→r4;M[2,N-1]→r5;M[2,N]→r6
109: r4-((r2-Q)/(r2-r1))(r4-r3)→S
110: if r5-r6>=180;r6+360→r6
111: if r6-r5>=180;r5+360→r5
112: r6-(r2-Q)(r6-r5)/(r2-r1)→A
113: if S<0;A+180→A;abs(S)→S
114: if A>=360;A-360→A
115: if A<0;A+360→A
116: rrnd(A/5,0)*5→A;rrnd(S,0)→S;S+100A→A
117: str(A)→C$;C$[2,6]→A$[11,15]
118: if A<10000;"0"→A$[11,11];C$[2,5]→A$[12,15]
119: if A<1000;"00"→A$[11,12];C$[2,4]→A$[13,15]
120: if A<100;"000"→A$[11,13];C$[2,3]→A$[14,15]
121: if A<10;"0000"→A$[11,14];C$[2,2]→A$[15,15]
122: if A=0;"000"→A$[11,13]
123: ret
124: "HEIGHT & TEMP & DEW POINT DEPRESSION":
125: rrnd(F[F,6]+val(P$[19,21])*10,0)→C
126: str(C)→C$;"000"→A$[3,5]
127: if D<=500;gto +5
128: if C>0 and C<10;C$[2,2]→A$[5,5]
129: if C>9 and C<100;C$[2,3]→A$[4,5]
130: if C>99 and C<1000;C$[2,4]→A$[3,5]
131: if C>999;C$[3,5]→A$[3,5];gto +3
132: if C>999 and C<10000;C$[2,4]→A$[3,5]
133: if C>9999;C$[3,5]→A$[3,5]
134: gsb "TEMP"
135: gsb "DP"
136: rrnd(F[F,6]+val(P$[19,21])*10,0)→Q
137: gsb "ANGLES"
138: ret
139: "R→P":√(XX+YY)→F
140: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
141: if A<0;360+A→A
142: ret
143: "P→R":F*cos(A)→X;F*sin(A)→Y
144: ret
145: if not flg3;gto +207
146: 10→C[1]
147: "METV"→A$[1,4];P$[6,9]→A$[5,8]
148: D$[4,5]→A$[9,10];P$[10,12]→A$[11,13];D$[6,7]→A$[14,15]
149: gsb "ARRAY 15"
*23470
```

```
150: P$[13,14]→A$[1,2];P$[19,21]→A$[3,5];gsb "ARRAY 5"
151: 10→C[1]
152: gto +8
153: "SUBHEAD":              "→A$[6,15]
154: val(D$[1,2])→Z;str(prnd(Z+50,0))→A$[6,8]
155: A$[7,8]→A$[6,7]
156: str(prnd(val(D$[4,7]).01,0))→A$[8,10];A$[9,10]→A$[8,9]
157: if val(A$[8,9])<10;A$[8,8]→A$[9,9];"0"→A$[8,8]
158: C$[1,5]→A$[11,15]
159: ret
160: 10val(P$[19,21])→M[1,1];H[-2,5]→M[2,1];H[-2,6]→M[3,1]
161: H[-1,5]→A;H[-1,6]→B;gsb "P→R"
162: X→r1;Y→r2
163: .0001→H[0,4]
164: for P=1 to 45;if H[P,4]=0;gto +11
165: P-1→J
166: 6367650→S
167: √((S+H[P,1])^2-S^2cos(H[P,6])^2)-Ssin(H[P,6])→r3
168: (Scos(H[P,6])/(S+H[P,1]))r3→B
169: H[P,5]→A;gsb "P→R"
170: r1→r4;X→r1;r4-r1→X;r2→r5;Y→r2;r5-r2→Y;gsb "P→P"
171: (P/(H[P,4]-H[J,4])).03238→S→M[3,P+1]
172: A→M[2,P+1]
173: H[J,1]+(H[P,1]-H[J,1])/2+10val(P$[19,21])→H→M[1,P+1]
174: next P
175: "1000085007000500040003000250020001500100"→W$
176: spc 1;" TTAA"→A$[1,5];gsb "SUBHEAD"
177: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
178: M[1,N]→D[5]
179: for N=2 to Y[1];if F[N,6]<D[5];next N
180: "/"→A$[10,10]
181: for W=9 to 0 by -1
182: if F[N-1,4]<=val(W$[W4+1,W4+4]);W$[W4+2,W4+2]→A$[10,10];gto +2
183: next W
184: gsl "ARRAY 15"
185: "/////////////////"→A$
186: 0→E;gsb "HEIGHT & TEMP & DEW POINT DEPRESSION"
187: "90"→A$[1,2];str(F[0,4])→C$;C$[2,4]→A$[3,5]
188: if F[0,4]>1000;C$[3,5]→A$[3,5]
189: H[-2,5]→A;H[-2,6]→S;prnd(A/5,0)*5→A
190: prnd(S,0)→S;S+100A→A;str(A)→A$[11];A$[12,16]→A$[11,15]
191: if A<10000;A$[11,14]→A$[12,15];"0"→A$[11,11]
192: if A<1000;A$[11,14]→A$[12,15];"0"→A$[11,11]
193: if A=0;"000"→A$[11,13]
194: A$[11,15]→Z$[1,5]
195: gsl "ARRAY 15"
196: 0→N
197: for P=1 to Y[1]
198: val(W$[W4+1,W4+4])→E
199: "///////////////"→A$
*23417
```

```
200: if B=1000 and C$[10,10]#"N";Q$[10,12]→A$[3,5];gto +5
201: if B=850 and C$[6,6]#"N";Q$[6,8]→A$[3,5];gto +4
202: if B<F[F,4] and B#F[F,4];gto +7
203: if B=F[-1,3] or B=F[-1,4] or B=F[-1,5] or B=F[-1,6];gto +2
204: gsl "HEIGHT & TEMP & DEW POINT DEPRESSION"
205: W$[W4+2,W4+3]→A$[1,2]
206: gsl "ARRAY 15"
207: if B#100;W+1→W
208: if B=100;A$[3,5]→Z$[6,8];gto +2
209: next F
210: if F[Y[1],4]>200;gto +8
211: for L=1 to Y[1];if F[L,4]>500;next L
212: L+1→N
213: if N>Y[1];gto +5
214: prnd((F[L,3]-F[N,3])/.001(F[N,6]-F[L,6]),-1)→C
215: if C<=2 and F[N,6]-F[L,6]>=2000;gto +4
216: if C<=2 and F[N,6]-F[L,6]<2000;N+1→N;gto -3
217: next L
218: gto +13
219: F[L,4]→O;F[L,6]→H;L→F
220: "////////////////"→A$
221: gsb "TEMP"
222: gsb "DP"
223: H+10val(P$[19,21])→Q;gsb "ANGLES"
224: "88"→A$[1,2];str(O)→C$;C$[2,4]→A$[3,5]
225: if val(A$[3,5])<100;gto +3
226: gsb "ARRAY 15"
227: gto +5
228: val(A$[1,5])→A[1];val(A$[11,15])→A[3]
229: if A$[9,10]="//";val(A$[6,8])→A[2]
230: if A$[9,10]#"//";val(A$[6,10])→A[2];gto +2
231: "88999"→A$;gsb "ARRAY 5"
232: if F[Y[1],4]>500;gto +27
233: M[3,1]→V
234: for N=2 to 46;if H[N-1,4]=0;gto +3
235: if M[3,N]>V;M[3,N]→V;M[2,N]→O;N→C
236: next N
237: if V<60;gto +21
238: 1000/(M[1,C]-M[1,C-1])→r13;1000/(M[1,C+1]-M[1,C])→r12
239: abs(V-M[3,C-1])r13→r13;abs(V-M[3,C+1])r12→r12
240: for L=1 to 60;if F[L,6]<=M[1,C] and F[L+1,6]>M[1,C];gto +2
241: next L
242: if F[L,4]>500;gto +16
243: F[L,4]+(M[1,C]-F[L,6])(F[L+1,4]-F[L,4])/(F[L+1,6]-F[L,6])→P
244: "77"→A$[1,2];P→O;if P<100;10P→O
245: gsb "STRING"
246: prnd(O/5,0)5→O;prnd(V,0)→V;1000+V→V
247: str(V)→C$;C$[2,6]→A$[6,10];"4"→A$[11,11]
248: if r13<1;100+prnd(r12,0)→r13;str(r13)→C$;C$[2,5]→A$[12,15]
249: 100prnd(r13,0)+prnd(r12,0)→r13;str(r13)→C$;C$[2,5]→A$[12,15]
*6390
```

```
250: if r13<1000;"0"→A$[12,12];C$[2,4]→A$[13,15]
251: if B[C+1,4]=0;"//"→A$[14,15]
252: if P<100;gto +3
253: gsb "ARRAY 15"
254: gto +5
255: val(A$[1,5])→A[4];val(A$[6,10])→A[5]
256: if A$[14,15]="//";val(A$[11,13])→A[6]
257: if A$[14,15]#"//";val(A$[6,10])→A[6]
258: "77999"→A$;gsb "ARRAY 5"
259: gsb "FOOT"
260: gto 1;" TTBB"→A$[1,5];gsb "SUBHEAD"
261: "/"→A$[10,10];gsb "ARRAY 15"
262: 1→F[11];0→W;F[0,1]→B[3]→B[4];F[0,2]→B[5]→B[6]
263: for M=1 to Y[1]
264: if prnd(F[M,1],-1)>prnd(B[3],-1);F[M,1]→B[3];M→B[7]
265: if prnd(F[M,1],-1)<prnd(B[4],-1);F[M,1]→B[4];M→B[8]
266: if prnd(F[M,2],0)>prnd(B[5],0);F[M,2]→B[5];M→B[9]
267: if prnd(F[M,2],0)<prnd(B[6],0) and F[M,2]#0;F[M,2]→B[6];M→B[10]
268: if F[M,2]=0 and B[11]#0;0→B[11];M-1→B[12]
269: next M
270: gto +8
271: "FOUND":Y[1]→B[2]
272: if B[7]>B[1] and B[7]<B[2];B[7]→B[2]
273: if B[8]>B[1] and B[8]<B[2];B[8]→B[2]
274: if B[9]>B[1] and B[9]<B[2];B[9]→B[2]
275: if B[10]>B[1] and B[10]<B[2];B[10]→B[2]
276: if B[12]>B[1] and B[12]<B[2];B[12]→B[2]
277: ret
278: 0→I[1]→P;gto +20
279: 0→I[3]→A→B[4]
280: for M=B[1]+1 to B[2]
281: (F[M,6]-F[B[1],6])/(F[B[2],6]-F[B[1],6])→C
282: F[M,1]-F[B[1],1]-C(F[B[2],1]-F[B[1],1])→A
283: if prnd(F[M,1],-1)=prnd(F[B[7],1],-1);M→B[2];gto +5
284: if prnd(F[M,1],-1)=prnd(F[B[8],1],-1);M→B[2];gto +4
285: if abs(A)>1 and F[M,4]>=300 and abs(A)>B[4];M→B[3];abs(A)→B[4]
286: if abs(A)>2 and F[M,4]<300 and abs(A)>B[4];M→B[3];abs(A)→B[4]
287: next M
288: if B[3]>B[1] and B[3]<B[2];B[3]→B[2];gto -9
289: 0→B[3]→A;if B[1]>B[12];gto +8
290: for M=B[1]+1 to B[2]
291: (F[M,6]-F[B[1],6])/(F[B[2],6]-F[B[1],6])→C
292: F[M,2]-F[B[1],2]-C(F[B[2],2]-F[B[1],2])→A
293: if prnd(F[M,2],0)=prnd(F[B[9],2],0);M→B[2];gto +4
294: if prnd(F[M,2],0)=prnd(F[B[10],2],0);M→B[2];gto +3
295: if abs(A)>10 and F[M,2]>20;M→B[2];gto +2
296: next M
297: B[2]→P
298: if F[P,4]<100;gto +13
299: "//////////"→A$[1,10];if D>9;1→D
*4276
```

```
300: str(prnd(D11,0))→A$[1,3];A$[2,3]→A$[1,2];"/"→A$[3,3]
301: if D=0;"00"→A$[1,2]
302: prnd(F[F,4],0)→Q
303: gsb "STRING"
304: gsb "TEMP"
305: gsb "DP"
306: 1+D→D
307: gsb "ARRAY 10"
308: if F=Y[1];gto +3
309: F→Y[1];gsb "BOUND"
310: gto -31
311: gsb "FOOT"
312: spc 1;" PPBP"→A$[1,5];gsb "SUBHEAD"
313: "0"→A$[10,10];gsb "ARRAY 15"
314: "010203040607080912141620253035507 0"→W$[1,34]
315: int(val(P$[19,21])*.03281+1)→r10
316: int((F[Y[1],6]/10+val(P$[19,21]))*.03281)→r9
317: if F[Y[1],4]>100;gto +2
318: int((val(Z$[6,8])+val(P$[19,21])+1000)*.03281)→r9
319: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
320: if int(M[1,N]*.003281)<r9;int(M[1,N]*.003281)→r9
321: "90//"→I$[1,5];2→D
322: "                    "→F$[1,15];Z$[1,5]→F$[1,5]
323: for W=0 to 16
324: val(W$[W2+1,W2+2])→Q
325: if r10>Q;next W
326: if Q>r9;gto +13
327: Q*304.8→Q
328: gsb "ANGLES"
329: if D=1;"9////"→I$[1,5];"                    "→F$[1,15]
330: if D=1;W$[W2+1,W2+2]→I$[2,3];A$[11,15]→F$[1,5]
331: if D=2;W$[W2+2,W2+2]→I$[4,4];A$[11,15]→F$[6,10]
332: if D=3;W$[W2+2,W2+2]→I$[5,5];A$[11,15]→F$[11,15]
333: if D#3 and W$[W2+1,W2+1]=W$[W2+3,W2+3];D+1→D;next W
334: str(r9)→B$;if r9>=10 and B$[2,2]=W$[W2+1,W2+1];D+1→D;next W
335: if r9<10 and W$[W2+1,W2+1]="0";D+1→D;next W
336: I$[1,5]→A$[1,5];gsb "ARRAY 5"
337: F$[1,15]→A$[1,15];gsb "ARRAY 15"
338: 1→I;next W
339: r9*304.8→Q;gsb "ANGLES"
340: if r9=val(W$[W2-1,W2]) and D=1;gto +10
341: if r9=val(W$[W2-1,W2]) and D#1;gto +7
342: str(r9)→B$;if r9<10;B$[2,2]→B$[3,3];"0"→B$[2,2]
343: if W$[W2-1,W2-1]#B$[2,2];1→D
344: if D=1;"9////"→I$[1,5];"                    "→F$[1,15]
345: if D=1;B$[2,3]→I$[2,3];A$[11,15]→F$[1,5]
346: if D=2;B$[3,3]→I$[4,4];A$[11,15]→F$[6,10]
347: if D=3;B$[3,3]→I$[5,5];A$[11,15]→F$[11,15]
348: I$[1,5]→A$[1,5];gsb "ARRAY 5"
349: F$[1,15]→A$[1,15];gsb "ARRAY 15"
*12315
```

```
350: csb "FOC1"
351: if F[Y[1],4]<100;trk 1;ldf 11
352: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
353: trl 0;ldp 0,0,6
354: end
*31475
```

(4) AWS Output Second Transmission

TRACK 1 FILE 11

```
0: dsp "USAF AWS MESSAGE"
1: gto +113
2: "A→B":fmt b,z;par 0
3: for Q=1 to r14
4: for K=1 to 2;pos(H$[K],C$[Q,Q])→J;if J;sfg K;gto +2
5: next K
6: if J=1 or J=3 or J=5 or J=9;cfg 1,2,5,6;gto +5
7: if flg1 and flg5;cfg 1;gto +4
8: if flg1;wtc 2,2;wrt 2,27;wtc 2,0;sfg 5;cfg 1,6;gto +3
9: if flg2 and flg6;cfg 2;gto +2
10: sfg 6;cfg 2,5;wtc 2,2;wrt 2,31;wtc 2,0
11: wtc 2,2;wrt 2,J-1;wtc 2,0
12: next Q
13: ret
14: "FOOT":if flg4 and flg7;par 0;fmt 2,b,z;wtc 2,2;wrt 2.2,15;wtc 2,0
15: fmt 2,z,cl
16: if flg4 and flg8;par 2;wtc 2,2;wrt 2.2,char(59);wtc 2,0
17: dsp ";";prt ";"
18: 10→C[1]
19: ret
20: "DELETE":cfg 3;dsp "MESSAGE BEING DELETED";wait 1000;ret
21: "ARRAY 15":
22: "        "→C$[1,6]
23: A$[1,5]→C$[2,6];gsb "ARRAY 6"
24: A$[6,10]→C$[2,6];gsb "ARRAY 6"
25: A$[11,15]→C$[2,6];gsb "ARRAY 6"
26: dsp A$[1,15];prt A$[1,15]
27: ret
28: "ARRAY 10":
29: "        "→C$[1,6]
30: A$[1,5]→C$[2,6];gsb "ARRAY 6"
31: A$[6,10]→C$[2,6];gsb "ARRAY 6"
32: dsp A$[1,10];prt A$[1,10]
33: ret
34: "ARRAY 5":
35: "        "→C$[1,6]
36: A$[1,5]→C$[2,6];gsb "ARRAY 6"
37: dsp A$[1,5];prt A$[1,5]
38: ret
39: "ARRAY 6":
40: if C$[1,6]="        ";ret
41: C[1]+1→C[1];if C[1]<10;gto +8
42: par 2
43: if flg4 and flg8;fmt 4,z,b;wtc 2,2;wrt 2.4,141,10;wtc 2,0
44: if flg4 and flg7;fmt 4,b,z;par 0;wtc 2,2;wrt 2.4,2,8;wtc 2,0
45: 0→C[1];C$[2,6]→C$[1,5]
46: if flg4 and flg7;5→r14;gsb "A→B"
47: if flg4 and flg8;par 2;fmt 4,z,c5;wtc 2,2;wrt 2.4,C$[1,5];wtc 2,0
48: gto +3
49: if flg4 and flg8;par 2;fmt 4,z,c6;wtc 2,2;wrt 2.4,C$[1,6];wtc 2,0
*26909
```

```
50: if flg4 and flg7;6→r14;gsb "A→B"
51: "        "→C$[1,6]
52: ret
53: "STRING":if Q<1000 and Q>99;str(Q)→A$[3,6];A$[4,6]→A$[3,5]
54: if Q<100 and Q>9;str(Q)→A$[3,5];"0"→A$[3,3]
55: ret
56: "TEMP":prnd(F[F,1],-1)→T;frc(T)→Q
57: if sgn(T)=1 or sgn(T)=0;if Q=.1 or Q=.3 or Q=.5 or Q=.7 or Q=.9;T-.1→T
58: if sgn(T)=-1;if Q=.2 or Q=.4 or Q=.6 or Q=.8 or Q=0;T-.1→T
59: abs(T)→T
60: "000"→A$[6,8];str(10T)→C$
61: if T>0 and T<1;C$[2,2]→A$[8,8]
62: if T>.9 and T<10;C$[2,3]→A$[7,8]
63: if T>9.9;C$[2,4]→A$[6,8]
64: ret
65: "DP":if prnd(F[F,1],-1)<-40;"//"→A$[9,10];ret
66: if F[F,2]=0;2→F[F,2]
67: (F[F,2]/100)6.11tn^(7.5F[F,1]/(237.3+F[F,1]))→E
68: (237.3log(E)-186.527)/(8.286-log(E))→E
69: prnd(F[F,1]-E,-1)→E
70: "00"→A$[9,10]
71: if E>0 and E<1;str(prnd(10E,0))→C$;C$[2,2]→A$[10,10]
72: if E>.9 and E<5;str(prnd(10E,0))→C$;C$[2,3]→A$[9,10]
73: if E>4.9;str(prnd(50+E,0))→C$;C$[2,3]→A$[9,10]
74: ret
75: "ANGLES":for N=2 to 44;if Q>M[1,N];next N
76: if M[3,N]=0 and M[2,N]=0;"        "→A$[11,15];ret
77: M[1,N-1]→r1;M[1,N]→r2;M[3,N-1]→r3;M[3,N]→r4;M[2,N-1]→r5;M[2,N]→r6
78: r4-((r2-Q)/(r2-r1))(r4-r3)→S
79: if r5-r6>=180;r6+360→r6
80: if r6-r5>=180;r5+360→r5
81: r6-(r2-Q)(r6-r5)/(r2-r1)→A
82: if S<0;A+180→A;abs(S)→S
83: if A>=360;A-360→A
84: if A<0;A+360→A
85: prnd(A/5,0)*5→A;prnd(S,0)→S;S+100A→A
86: str(A)→C$;C$[2,6]→A$[11,15]
87: if A<10000;"0"→A$[11,11];C$[2,5]→A$[12,15]
88: if A<1000;"00"→A$[11,12];C$[2,4]→A$[13,15]
89: if A<100;"000"→A$[11,13];C$[2,3]→A$[14,15]
90: if A<10;"0000"→A$[11,14];C$[2,2]→A$[15,15]
91: if A=0;"000"→A$[11,13]
92: ret
93: "HEIGHT & TEMP & DEW POINT DEPRESSION":
94: prnd(F[F,6]+val(P$[19,21])*10,0)→Q
95: str(Q)→C$;C$[3,5]→A$[3,5]
96: gsb "TEMP"
97: gsb "DP"
98: prnd(F[F,6]+val(P$[19,21])*10,0)→Q
99: gsb "ANGLES"
*8375
```

```
100: ret
101: "R→P":√(XX+YY)→R
102: atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
103: if A<0;360+A→A
104: ret
105: "P→R":R*cos(A)→X;R*sin(A)→Y
106: ret
107: "SUBHEAD":"              "→A$[6,15]
108: val(D$[1,2])→Z;str(prnd(Z+50,0))→A$[6,8]
109: A$[7,8]→A$[6,7]
110: str(prnd(val(D$[4,7]).01,0))→A$[8,10];A$[9,10]→A$[8,9]
111: if val(A$[8,9])<10;A$[8,8]→A$[9,9];"0"→A$[8,8]
112: Q$[1,5]→A$[11,15]
113: ret
114: "7050302010070503020100000000000000000000"→W$
115: spc 2;" TTCC"→A$[1,5];gsb "SUBHEAD"
116: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
117: M[1,N]→B[5]
118: for N=2 to Y[1];if F[N,6]<B[5];next N
119: for W=4 to 0 by -1
120: if F[N-1,4]<=val(W$[W2+1,W2+2]);W$[W2+1,W2+1]→A$[10,10];gto +2
121: next W
122: gsb "ARRAY 15"
123: "////////////////"→A$
124: 0→F;gsb "HEIGHT & TEMP & DEW POINT DEPRESSION"
125: 0→W
126: for F=1 to Y[1]
127: val(W$[W2+1,W2+2])→R
128: "////////////////"→A$
129: if B<F[F,4] and B#F[F,4];gto +6
130: if B=F[-1,3] or B=F[-1,4] or B=F[-1,5] or B=F[-1,6];gto +2
131: gsb "HEIGHT & TEMP & DEW POINT DEPRESSION"
132: W$[W2+1,W2+2]→A$[1,2]
133: gsb "ARRAY 15"
134: if B#1;W+1→W
135: next F
136: if A[1]=0;gto +7
137: str(A[1])→C$;C$[2,6]→A$[1,5];str(A[3])→C$;C$[2,6]→A$[11,15]
138: str(A[2])→C$
139: if A[2]>=1000;C$[2,6]→A$[6,10]
140: if A[2]<1000;C$[2,4]→A$[6,8];"//"→A$[9,10]
141: gsb "ARRAY 15"
142: gto +2
143: "88999"→A$;gsb "ARRAY 5"
144: if A[4]=0;gto +7
145: str(A[4])→C$;C$[2,6]→A$[1,5];str(A[5])→C$;C$[2,6]→A$[6,10]
146: str(A[6])→C$
147: if A[6]<1000;C$[2,4]→A$[11,13];"//"→A$[14,15]
148: if A[6]>=1000;C$[2,6]→A$[11,15]
149: gsb "ARRAY 15"
*28487
```

```
150: gto +2
151: "77999"→A$;gsb "ARRAY 5"
152: gsb "ROOT"
153: src 1;" TTLD"→A$[1,5];gsb "SUBHEAD"
154: "/"→A$[10,10];gsb "ARRAY 15"
155: 0→V;1→D
156: gto +9
157: "BOUND":Y[1]→B[2]
158: if B[7]>B[1] and B[7]<B[2];B[7]→B[2]
159: if B[8]>B[1] and B[8]<B[2];B[8]→B[2]
160: if B[9]>B[1] and B[9]<B[2];B[9]→B[2]
161: if B[10]>B[1] and B[10]<B[2];B[10]→B[2]
162: if B[12]>B[1] and B[12]<B[2];B[12]→B[2]
163: if B[B[2],4]>=100;B[2]→B[1];gto -6
164: rot
165: 0→B[1]→F;gsb "BOUND"
166: 0→B[3]→A→B[4]
167: for N=B[1]+1 to B[2]
168: (B[N,6]-B[B[1],6])/(B[B[2],6]-B[B[1],6])→C
169: B[N,1]-B[B[1],1]-C(B[B[2],1]-B[B[1],1])→A
170: if rnd(B[N,1],-1)=rnd(B[7],-1);N→B[2];gto +4
171: if rnd(B[N,1],-1)=rnd(B[8],-1);N→B[2];gto +3
172: if abs(A)>2 and abs(A)>B[4];N→B[3];abs(A)→B[4]
173: next N
174: if B[3]>B[1] and B[3]<B[2];B[3]→B[2];gto -8
175: 0→B[3]→A;if B[1]>B[12];gto +8
176: for N=B[1]+1 to B[2]
177: (B[N,6]-B[B[1],6])/(B[B[2],6]-B[B[1],6])→C
178: B[N,2]-B[B[1],2]-C(B[B[2],2]-B[B[1],2])→A
179: if rnd(B[N,2],0)=rnd(B[9],0);N→B[2];gto +4
180: if rnd(B[N,2],0)=rnd(B[10],0);N→B[2];gto +3
181: if abs(A)>10 and B[N,2]>20;N→B[2];gto +2
182: next N
183: B[2]→F
184: if B[F,4]<1;gto +13
185: if B[F,4]>=100;gto +9
186: "//////////"→A$[1,10];if D>9;1→D
187: str(rnd(D11,0))→A$[1,3];A$[2,3]→A$[1,2];"/"→A$[3,3]
188: 10rnd(B[F,4],-1)→G
189: gsb "STPINC"
190: gsb "TEMP"
191: gsb "DP"
192: 1+D→D
193: gsb "ARRAY 10"
194: if F=Y[1];gto +3
195: F→B[1];gsb "BOUND"
196: gto -30
197: gsb "ROOT"
198: "035050070090100110140150160170"→V$[1,30]
199: src 1;" PPDD"→A$[1,5];gsb "SUBHEAD"
*11]36
```

```
200: "0"→A$[10,10];gsb "ARRAY 15"
201: int((E[Y[1],6]/10+val(P$[19,21]))*.03281)→r9
202: for N=45 to 1 by -1;if M[2,N]=0 and M[3,N]=0;next N
203: if int(M[1,N]*.003281)<r9;int(M[1,N]*.003281)→r9
204: int((val(Z$[6,8])+val(P$[19,21])+1000)*.03281)→r10
205: 1→D
206: for W=0 to 9
207: val(W$[W3+1,W3+3])→Q
208: if r10>Q;next W
209: if Q>r9;gto +12
210: Q*304.8→Q
211: gsb "ANGLES"
212: "9////"→I$[1,5];"                    "→F$[1,15]
213: if Q>=100*304.8;"1"→I$[1,1]
214: W$[W3+2,W3+3]→I$[2,3];A$[11,15]→F$[1,5]
215: str(r9)→B$;if r9<100 and B$[2,2]=W$[W3+2,W3+2];2→D;next W
216: if r9>=100 and B$[3,3]=W$[W3+2,W3+2];2→D;next W
217: I$[1,5]→A$[1,5];gsb "ARRAY 5"
218: F$[1,5]→A$[1,5];gsb "ARRAY 5"
219: next W
220: gto +10
221: r9*304.8→Q;gsb "ANGLES"
222: if r9=val(W$[W3-2,W3]);gto +8
223: str(r9)→B$;if r9<100;B$[2,3]→B$[3,4]
224: if D=1;"9////"→I$[1,5];"            "→F$[1,10]
225: if D=1 and Q>=100*304.8;"1"→I$[1,1]
226: if D=1;B$[3,4]→I$[2,3];A$[11,15]→F$[1,5]
227: if D=2;B$[4,4]→I$[4,4];A$[11,15]→F$[6,10]
228: I$[1,5]→A$[1,5];gsb "ARRAY 5"
229: F$[1,10]→A$[1,10];gsb "ARRAY 10"
230: gsb "FOOT"
231: if flg4;wtc 2,2;for N=1 to 75;wtb 2,0;next N;wtc 2,0
232: trk 0;ldp 0,0,6
233: end
*13212
```

f. Flight Information

TRACK 1 FILE 9

```
0: cfg 0;sfg 3;dsp "PROCESSING FLIGHT INFORMATION";gto +8
1: "I":spc 1;fxd 0;prt "SIG LEVEL",I;fxd 1;prt " TIME",F[I,5]
2: fxd 1;prt " PRESS,mb",F[I,4];prt "MISSING MET DATA";spc 1;ret
3: "DELETE":cfg 3;dsp "DELETING FLIGHT INFORMATION";ret
4: "R→P":√(XX+YY)→R;atn(Y/(X+1e-90*(X=0)))+2*sgn(Y)*atn(1e99)*(X<0)→A
5: if A<0;360+A→A
6: ret
7: "P→R":R*cos(A)→X;R*sin(A)→Y;ret
8: dim F[-1:60,1:6],Y[2]
9: trk 0;ldf 10,F[*],Y[*]
10: spc 5;prt "****************","FLT INFORMATION";spc 2
11: for T=Y[1] to 1 by -1;if F[T,3]=0;next T
12: spc 1;fxd 0;prt "SIG LEVEL",T
13: fxd 1;prt " TIME,m",F[T,5]
14: fxd 0;prt " HEIGHT,m",F[T,6]
15: fxd 1;prt " TEMP,C",F[T,1]
16: fxd 0;prt " HUM,%",F[T,2]
17: fxd 1;prt " PRESS,mb",F[T,4]
18: fxd 1;prt " TEMP,Tv",F[T,3]
19: for I=T-1 to 0 by -1;if F[I,3]=0;gsb "I"
20: if F[I,3]=0;next I
21: prnd((F[T,3]-F[I,3])/.001(F[T,6]-F[I,6]),-1)→G
22: fxd 1;prt " LAPSE,C",G
23: (F[T,6]-F[I,6])/(F[T,5]-F[I,5])→G
24: fxd 0;prt " RATE/RISE",G
25: F[T,6]/F[T,5]→G
26: fxd 0;prt " AVE-RISE",G
27: if not flg3;gto +7
28: next T
29: spc 2;fxd 0;prt "SURFACE"
30: fxd 1;prt " TEMP,C",F[0,1]
31: fxd 0;prt " HUM,%",F[0,2]
32: fxd 1;prt " PRESS,mb",F[0,4]
33: fxd 1;prt " TEMP,Tv",F[0,3]
34: dim P$[102],Q$[16],O$[16],L$[16],N$[6],G$[192],D$[16]
35: trk 0;ldf 9,P$,Q$,O$,L$,N$,G$,D$
36: dim H[-2:45,1:6]
37: ldf 11,H[*]
38: sfg 3;spc 5;prt "ZONE INFORMATION"
39: if flg0;spc 3;fxd 0;prt "SURFACE"
40: if flg0;prt " ALTgeom",H[0,1]
41: if flg0;fxd 3;prt " TIME",H[0,4]
42: if flg0;fxd 0;prt " HOR DST",H[-1,6]
43: if flg0;fxd 3;prt " AZ OFF",H[-1,5]
44: if flg0;fxd 0;prt " ALTmsl",val(P$[19,21])*10
45: if flg0;fxd 0;prt " WDIR,deg",H[-2,5]
46: if flg0;fxd 0;prt " WSPEED,k",H[-2,6]
47: H[-1,5]→A;H[-1,6]→R;gsb "P→R"
48: X→r1;Y→r2
49: .00001→H[0,4]
*27494
```

```
50: for N=0 to 45;if H[N,4]=0;gto +20
51: N-1→J
52: 6367650→S
53: √((S+H[N,1])^2-S^2cos(H[N,6])^2)-Ssin(H[N,6])→r3
54: (Scos(H[N,6])/(S+H[N,1]))r3→R
55: H[N,5]→A;gsb "P→P"
56: r1→r4;X→r1;r4-r1→X;r2→r5;Y→r2;r5-r2→Y;gsb "P→P"
57: (R/(H[N,4]-H[J,4])).03238→S
58: H[J,1]+(H[N,1]-H[J,1])/2+val(P$[19,21])*10→R
59: if flg0;spc 1;fxd 0;prt "ALTmsl",R
60: if flg0;fxd 0;prt "WDIR,deg",A
61: if flg0;fxd 0;prt "WSPEED,k",S
62: spc 1;fxd 0;prt "ZONE",N
63: prt " ALTgeom",H[N,1]
64: fxd 3;prt " TIME",H[N,4]
65: fxd 3;prt " ELEV",H[N,6]
66: if N=0;fxd 0;prt " HOR DST",H[-1,6]
67: fxd 3;prt " AZ",H[N,5]
68: if not flg3;gto +2
69: next N
70: prt "****************";spc 5
71: dsp "PROCESSING FLIGHT CONTINUATION"
72: trk 0;ldp 6
73: end
*3668
```

c. B. Tape

TRACK 1 FILE 2

```
0: sfg 1
1: dim U$[1260],B$[5],A$[180],E$[7],C$[32],D$[17]
2: dim W$[56],Z$[895],V$[96],B[25]
3: dim X$[1],F$[39]
4: if flg1;gto +5
5: for N=1 to 895;"0"→Z$[N];next N
6: "IS BENEX OPERATIONAL?"→C$;gsb "Y/N"
7: if A=-2tn^10;cfg 1;gto +5
8: if A=-1tn^10;sfg 1
9: dsp "PREPARATION OF PRESS TAPE";gsb "BBB"
10: spc 5;time 500
11: prt "Use ASCII tape  one-inch wide..","TAPE FORMAT: NWS";spc 2
12: prt "PRESS TAPE ENTRY","----------------"
13: "        "→U$[1,7]
14: "99999                    0      "→F$[1,30]
15: "180  0  0        "→D$[1,17]
16: gto +61
17: "PRESS TAPE":2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep
18: if K<100;" "→B$[3,3];if K<10;"  "→B$[2,3]
19: fxd 0;dsp "ENTER PRESSURE AT CONTACT",B$[1,3];ent "",A$
20: if not flg1 and A$[1,2]="00";gto +103
21: if A$[1,7]#"-2tn^10";gto +3
22: dsp """DELETE"" CONT OR USE UPPER KEYS";gsb "BBB"
23: stp
24: if A#2;2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep;gto -5
25: "        "→E$[1,7]
26: if len(A$)>6;gto -9
27: fxd 1;val(A$[1,6])→L;fxd 0;prt A$[1,6],K;fxd 1;if L<10000;str(L)→E$[1,
28: " "→E$[1,1]
29: if L<1000;str(L)→E$[2,7];"  "→E$[1,2]
30: if L<100;str(L)→E$[3,7];"   "→E$[1,3]
31: if L<10;str(L)→E$[4,7];"    "→E$[1,4]
32: if L<1;str(L)→E$[5,7];"     "→E$[1,5]
33: if L<0;gto -14
34: if L>1100;prt "---REPEATING----";gsb "BBB"
35: if L>1100;gto -16
36: if L=0;"      0"→E$[1,7]
37: ret
38: "DELETE":dsp "DELETING CONTACT & PRESSURE";prt "---DELETING-----"
39: gsb "BBB"
40: K-1→M;gto 100
41: ret
42: "DI":fxd 1;2→A;""→A$;"0"→A$[1,1];beep;wait 150;beep;dsp C$[1,32]
43: ent "",A$;if A$[1,1]="0" or A$[1,1]="1" or A$[1,1]="2";gto +9
44: if A$[1,1]="3" or A$[1,1]="4" or A$[1,1]="5" or A$[1,1]="6";gto +8
45: if A$[1,1]="7" or A$[1,1]="8" or A$[1,1]="9" or A$[1,2]=".0";gto +7
46: if A$[1,2]=".1" or A$[1,2]=".2" or A$[1,2]=".3" or A$[1,2]=".4";gto +6
47: if A$[1,2]=".5" or A$[1,2]=".6" or A$[1,2]=".7" or A$[1,2]=".8";gto +5
48: if A$[1,2]=".9";gto +4
49: if A$[1,7]="-2tn^10";trk 0;ldp 0,0,6
*17312
```

```
50: stj
51: gto -9
52: if A#2;gto -10
53: val(A$)→L;if L<0;gto -11
54: "                                    "→C$[1,32]
55: ret
56: "Y/N":2→A;beep;dsp C$[1,32];ent "",A
57: if A#-2tn^10 and A#-1tn^10;gto -1
58: "                                    "→C$[1,32]
59: ret
60: "VISUAL DIGITS":
61: wtc 2,2;fmt b,z
62: if X$="0";par 2;wrt 2,126;par 2;wrt 2,1;par 2;wrt 2,1;par 0;wrt 2,126
63: if X$="1";par 2;wrt 2,127
64: if X$="2";par 3;wrt 2,114;par 3;wrt 2,9;par 3;wrt 2,9;par 3;wrt 2,6
65: if X$="3";par 2;wrt 2,65;par 3;wrt 2,9;par 3;wrt 2,9;par 2;wrt 2,119
66: if X$="4";par 2;wrt 2,15;par 3;wrt 2,8;par 3;wrt 2,8;par 3;wrt 2,126
67: if X$="5";par 3;wrt 2,79;par 3;wrt 2,9;par 3;wrt 2,9;par 2;wrt 2,113
68: if X$="6";par 2;wrt 2,126;par 3;wrt 2,9;par 3;wrt 2,9;par 2;wrt 2,114
69: if X$="7";par 3;wrt 2,113;par 2;wrt 2,9;par 3;wrt 2,1;par 3;wrt 2,7
70: if X$="8";par 3;wrt 2,118;par 3;wrt 2,9;par 3;wrt 2,9;par 3;wrt 2,118
71: if X$="9";par 3;wrt 2,70;par 1;wrt 2,9;par 1;wrt 2,9;par 2;wrt 2,126
72: if X$=" ";par 2;wrt 2,24;par 2;wrt 2,24;par 2;wrt 2,24;wrt 2,24
73: if X$="O";par 2;wrt 2,126,127,67,67,127,126
74: if X$="L";par 2;wrt 2,127,127,64,64,64
75: par 0;wrt 2,0
76: ret
77: "ENTER SERIAL NUMBER (xxx-xxxx)"→C$;gsb "DI"
78: if A$[4,4]="-" and len(A$)=8;A$[1,8]→F$[8,15];gto +3
79: dsp "SEVEN DIGITS ARE NEEDED";gsb "BBB"
80: gto -3
81: "ENTER DETENT VALUE (.xx)          "→C$[1,32];gsb "DI"
82: 100L→K;str(K)→A$;if K<10000;A$[2,5]→F$[17,20]
83: if K<1000;A$[2,2]→F$[17,17];A$[3,4]→F$[19,20];" "→F$[18,18]
84: if K<100;A$[2,3]→F$[19,20];"  "→F$[17,18]
85: if K<10;A$[2,2]→F$[20,20];"   "→F$[17,19]
86: "SER NO: "→C$[1,8];F$[8,15]→C$[9,16];prt C$[1,16];fxd 2
87: "DETENT: "→C$[1,8];str(L)→C$[12,16];"    "→C$[9,12]
88: prt C$[1,16];spc 2
89: "ARE PRINTED CHART VALUES OK?    "→C$[1,32];gsb "Y/N"
90: if A=-2tn^10;prt "---REPEATING----";gto -13
91: " "→F$[11,11]
92: val(F$[8,10])+val(F$[12,15])+val(F$[17,20])→L;if L>10000;L-10000→L
93: str(L)→A$
94: if L>1;A$[2,2]→F$[30,30]
95: if L>10;A$[2,3]→F$[29,30]
96: if L>100;A$[2,4]→F$[28,30]
97: if L>1000;A$[2,5]→F$[27,30]
98: prt "PRESS        CONT"
99: 1→M
*16753
```

```
100: for K=M to 179
101: str(prnd(K,0))→B$[1,4];B$[2,4]→B$[1,3];gsb "PRESS TAPE"
102: E$[1,7]→U$[1+K7,7+K7]
103: E$[2,5]→Z$[1+5(K-1),4+5(K-1)]
104: E$[7,7]→Z$[5+5(K-1),5+5(K-1)]
105: if val(E$)<1000;"0"→Z$[1+5(K-1),1+5(K-1)]
106: if val(E$)<100;"0"→Z$[2+5(K-1),2+5(K-1)]
107: if val(E$)<10;"0"→Z$[3+5(K-1),3+5(K-1)]
108: if val(E$)<1;"0"→Z$[4+5(K-1),4+5(K-1)]
109: if K=1;gto +4
110: if val(U$[1+7K,7+7K])<=val(U$[1+(K-1)7,7+(K-1)7]);gto +3
111: K-1→M;dsp "---REPEATING----";gsb "BBB"
112: prt "---REPEATING----";gto -12
113: next K
114: fxd 1;for K=1 to 179;val(U$[1+7K,7+7K])+Z→Z;next K
115: Z-int(Z/10000)*10000→Z
116: str(Z)→A$
117: if Z>0;A$[2,3]→D$[16,17]
118: if Z>1;A$[2,4]→D$[15,17]
119: if Z>10;A$[2,5]→D$[14,17]
120: if Z>100;A$[2,6]→D$[13,17]
121: if Z>1000;A$[2,7]→D$[12,17];spc 2
122: "CHECKSUM "→C$[1,9];D$[12,17]→C$[10,15];prt C$[1,15];spc 2
123: trk 1;rcf 0,W$,Z$,V$,B[*]
124: if not flg1;trk 0;ldp 2,0,310
125: dsp "TURN-ON REMEX";gsb "BBB"
126: if not ios2;wait 5000;gto -1
127: wtc 2,2;for N=0 to 50;fmt b,z;wrt 2,0;next N;wtc 2,0
128: F$[8,8]→X$;gsb "VISUAL DIGITS"
129: F$[9,9]→X$;gsb "VISUAL DIGITS"
130: F$[10,10]→X$;gsb "VISUAL DIGITS"
131: " "→X$;gsb "VISUAL DIGITS"
132: F$[12,12]→X$;gsb "VISUAL DIGITS"
133: F$[13,13]→X$;gsb "VISUAL DIGITS"
134: F$[14,14]→X$;gsb "VISUAL DIGITS"
135: F$[15,15]→X$;gsb "VISUAL DIGITS"
136: wtc 2,2;for N=1 to 7;fmt b,z;wrt 2,0;next N;wtc 2,0
137: "O"→X$;gsb "VISUAL DIGITS"
138: "L"→X$;gsb "VISUAL DIGITS"
139: " "→X$;gsb "VISUAL DIGITS"
140: "1"→X$;gsb "VISUAL DIGITS"
141: "9"→X$;gsb "VISUAL DIGITS"
142: "2"→X$;gsb "VISUAL DIGITS"
143: wtc 2,2;for N=1 to 7;fmt b,z;wrt 2,0;next N;wtc 2,0
144: par 2;wtc 2,2;for N=1 to 49;fmt b,z;wrt 2,127;next N;wtc 2,0
145: wtc 2,2;fmt b;wrt 2,127;wtc 2,0
146: wtc 2,2;fmt b,z;wrt 2,127;wtc 2,0
147: wtc 2,2;par 2;wrt 2,F$[1,30];wtb 2,127;wtc 2,0
148: for N=0 to 17;str(N)→B$[1,3];if N<10;"0"→A$[1,1];B$[2,2]→A$[2,2]
149: if N>9;B$[2,3]→A$[1,2]
*31005
```

```
150: "("→A$[3,3]
151: U$[1+70N,63+70N]→A$[3,65]
152: U$[65+70N,70+70N]→A$[66,71];"("→A$[3,3]
153: wtc 2,2;par 2;wrt 2,A$;wtb 2,127;wtc 2,0
154: next N
155: wtc 2,2;par 2;wrt 2,L$[1,17];wtb 2,127;wtc 2,0
156: wtc 2,2;for N=1 to 50;frt 1,2;wrt 2,127;next N;wtc 2,0
157: wtc 2,2;for N=0 to 75;wtb 2,0;next N;wtc 2,0
158: dsp "PRESSURE TAPE COMPLETE";stc 2
159: trk 0;loc 0,0,6
160: "EEL":beep;wait 150;beep;wait 150;beep;wait 3000;ret
161: end
*16833
```

h. TACFIRE

TRACK 1 FILE 3

```
0: dsp "TACFIRE TEST TRIAL IN PROGRESS";wait 5000;spc 5
1: dsp "LOOP";prt "LOOP";wait 200
2: for N=48 to 57;dsp char(N);prt char(N);wait 200;next N
3: for N=65 to 90;dsp char(N);prt char(N);wait 200;next N
4: dsp "TACFIRE TEST TRIAL COMPLETED";spc 2
5: trk 0;ldp 0,0,6
6: end
*15794
```

7. REFERENCES

FM 6-15, Artillery Meteorology, Headquarters, Department of the Army, MAR 70.

FM 6-16, Tables for Artillery Meteorology, Headquarters, Department of the Army, MAY 61.

FMH-4, Radiosonde Code. Federal Meteorologist Handbook No. 4, JAN 72.

FMH-6, Upper Wind Code. Federal Meteorologist Handbook No. 6, JAN 72.

Program Documentation: Radiosonde Sounding Reduction Program for FADAC (M-18), NOV 67.

Draft: Simplified Approach to Reduction of Radiosonde Telemetry Data by Digital Processors. D. R. Veazey, M. D. Merrill, JAN 75, ASL, WSMR, New Mexico.

Automatic Processing of Radiosonde Observations, US Weather Bureau, OCT 62.

Equations and Multipurpose GMD RAWIN Program for Automatic Processing of any type of Radiosonde. Novella S. Billions, US Army Missile Command, Redstone Arsenal, AL, DEC 70.

HP 09825-90000. Operating and Programming Manual for the Hewlett Packard 9825A Calculator. 76

DEP TM 11-6660-205-10, Operator's Manual, Meteorological Data Processing Group OL-192(*), US Army Electronics Command, Fort Monmouth, NJ.