

Series Analyzed

QC
874.3
.U68
no. 62



**NOAA Western Region Computer Programs
and Problems NWS WRCP No. 62**

CLINEWS

Ray Stuyvesant

**WSO Missoula, Montana
May 1991**

**U.S. DEPARTMENT OF
COMMERCE**

/ National Oceanic and
Atmospheric Administration

/ National Weather
Service

PREFACE

This Western Region publication series is a subset of our Technical Memorandum series. This series will be devoted exclusively to the exchange of information on and documentation of computer programs and related subjects. This series was initiated because it did not seem appropriate to publish computer program papers as Technical Memoranda; yet, we wanted to share this type of information with all Western Region forecasters in a systematic way. Another reason was our concern that in the developing AFOS-era there would be unnecessary and wasteful duplication of effort in writing computer programs in National Weather Service (NWS). Documentation and exchange of ideas and programs envisioned in this series hopefully will reduce such duplication. We also believe that by publishing the programming work of our forecasters, we will stimulate others to use these programs or develop their own programs to take advantage of the computing capabilities AFOS makes available.

We solicit computer-oriented papers and computer programs from forecasters for us to publish in this series. Simple and short programs should not be prejudged as unsuitable.

The great potential of the AFOS-era is strongly related to local computer facilities permitting meteorologists to practice in a more scientific environment. It is our hope that this series will help in developing this potential into reality.

NOAA WESTERN REGION COMPUTER PROGRAMS AND PROBLEMS NWS WRCP

- 1 Standardized Format for Computer Series. Revised January 1984. (PB85 109668)
- 2 AFOS Crop and Soil Information Report Programs. Kenneth B. Mielke, July 1979. (PB85 110419)
- 3 Decoder for Significant Level Transmissions of Raobs. John A. Jannuzzi, August 1979. (PB85 109676)
- 4 Precipitable Water Estimate. Elizabeth Morse, October 1979.
- 5 Utah Recreational Temperature Program. Kenneth M. Labas, November 1979.
- 6 Normal Maximum/Minimum Temperature Program for Montana. Kenneth B. Mielke, December 1979. (PB85 112878)
- 7 Plotting of Ocean Wave Energy Spectral Data. John R. Zimmerman, December 1979. (PB85 112860)
- 8 Raob Plot and Analysis Routines. John A. Jannuzzi, January 1980.
- 9 The SWAB Program. Morris S. Webb, Jr., April 1980. (PB80 196041)
- 10 Flash-Flood Procedure. Donald P. Laurine and Ralph C. Hatch, April 1980. (PB80 298658)
- 11 Program to Forecast Probability of Summer Stratus in Seattle Using the Durst Objective Method. John R. Zimmerman, May 1980.
- 12 Probability of Sequences of Wet and Dry Days. Hazen H. Bedke, June 1980. (PB80 223340)
- 13 Automated Montana Hourly Weather Roundup. Joe L. Johnston, July 1980. (PB81 102576)
- 14 Lightning Activity Levels. Mark A. Mollner, July 1980. (PB81 108300)
- 15 Two FORTRAN Applications of Wind-Driven Ekman Water Transport Theory: Upwelling Index and Storm Tide. Kent S. Short, July 1980. (PB81 102568)
- 16 AFOS System Local Data Base Save and Rebuild Procedures or a Master Doomsday Program. Brian W. Finke, July 1980. (PB81 108342)
- 17 AFOS/RDOS Translator Subroutine. Morris S. Webb, Jr., August 1980. (PB81 108334)
- 18 AFOS Graphics Creation from FORTRAN. Alexander E. MacDonald, August 1980. (PB81 205304)
- 19 DATAKEY0/ Repair Program. Paul D. Tolleson, August 1980. (PB81 102543)
- 20 Contiguous File Transfer from the DPCM to the DCM. Paul D. Tolleson, September 1980. (PB81 128035)
- 21 Freezing Level Program. Kenneth B. Mielke, September 1980. (PB81 128043)
- 22 Radar Boreighting Verification Program. Thomas E. Adler, November 1980. (PB81 182677)
- 23 Accessing the AFOS Data Base. Matthew Peroutka, January 1981. (PB81 190266)
- 24 AFOS Work Processor. Morris S. Webb, Jr., February 1981. (PB81 210007)
- 25 Automated Weather Log for Terminal Forecasting. John A. Jannuzzi, February 1981. (PB81-210999)
- 26 Program to Computer Downwind Concentrations from a Toxic Spill. John R. Zimmerman, February 1981. (PB81 205296)
- 27 Animation of AFOS Graphics. James R. Fors, August 1987 (revision). (PB87 220109/AS)
- 28 AFOS Interactive Graphics. James R. Fors, Don Laurine, and Sandy MacDonald, April 1981. (PB85 110401)
- 29 Computer Programs for Aviation Forecast Transmission. Kenneth B. Mielke and Matthew R. Peroutka, May 1981. (PB85 110518)
- 30 AFOS Product Collective Program. Morris S. Webb, Jr., September 1981. (PB85 109841)
- 32 Automation of Hourly Aviation Observation Calculations. W. Paul Duval, October 1981. (PB85 109650)
- 33 Mesoscale Objective Analysis. Andrew J. Spry and Jeffrey L. Anderson, December 1981. (PB85 109825)
- 34 Orographic Snowfall Rate Model for Alta, Utah. Steven K. Todd and Glenn E. Rasch, December 1981. (PB85 109874)
- 35 F-6 Monthly Climatic Summary Program For AFOS. Peter G. Mueller, May 1982. (PB85 109858)
- 36 Soaring Forecast Program. D.S. Toronto and G. R. Lussky, Revised March 1986. (PB86 173523/AS)
- 37 Program to Work Up Climatic Summary Weather Service Forms (F-6, F-52). Peter G. Mueller, August 1982. (PB85 109866)
- 38 The Hovmoller Diagram. Pamela A. Hudadoff, September 1982. (PB85 112159)
- 39 850-Millibar Charts Derived from Surface Data. Jeffrey L. Anderson, December 1982. (PB85 112175)
- 40 AFOS Vector Graphic to Grid Point Program. James R. Fors, December 1982. (PB85 109544)
- 41 A Pilot Briefing Program for the Background Partition. Kenneth B. Mielke and Joe L. Johnston, March 1983. (PB85 109551)
- 42 AEV Local Verification for Aviation, Precipitation, and Temperature Programs: AV, REL, TEM. Timothy W. Barker, Revised September 1987. (PB88 115662/AS)
- 43 OBLOG. Nancy Larsen, December 1983. (PB85 109528)

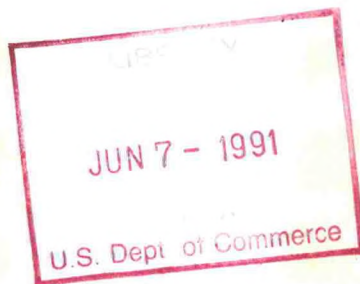
- 44 Communications Software for Olympics Micromation Computer System. Glen Sampson, June 1984. (PB85 109510)
- 45 PLOTFILE Appender. Wendy L. Wolf, July 1984. (PB85 109502)
- 46 Spectral Wave Data Analysis (Non-Directional). Lawrence Dunn, August 1984. (PB85 109577)
- 47 Isentropic Objective Analysis. Jeffrey L. Anderson, August 1984. (PB85 112167)
- 48 Hurricane Plotting Program. Paul D. Tolleson, October 1984. (PB85 121432)
- 49 Hemispheric Spectral Wave Analysis (Waves 0 to 7). Mary F. Milkovich, August 1985. (PB86 108719/AS)
- 50 AOS Graphic to Grid Point Conversion and Departure from Normal Programs. Jeffrey L. Anderson and Mark A. Mathewson, August 1985. (PB85 248110/AS)
- 51 Sunrise/Sunset and Moonrise/Moonset. Glenn R. Lussky, January 1986 (Revised). (PB86 157229/AS)
- 52 Objective Contour Analysis Using the Surface of Least Bending (Spline Analysis). Les Colin, November 1985. (PB86 128675/AS)
- 53 DATACOL - AFOSPLOT Program. Donald P. Laurine and Timothy K. Helble, February 1986. (PB86 161866/AS)
- 54 Hemispheric Spectral Analysis Program. Craig C. Peterson, April 1986. (PB 183662/AS)
- 55 Convective Cross Section Analysis. Timothy W. Barker, June 1987. (PB87 204566)
- 56 SWELL Program. Craig C. Peterson, August 1987. (PB87 229795/AS)
- 57 Watchdog Program. William R. Schneider and Craig C. Peterson, October 1988. (PB89 122535/AS)
- 58 Daily Climate Summary for MAPSO. Joe L. Johnston, August 1989. (PB89 230841/AS) - Revised May 1991
- 59 SEAPLOT. Bob Diaz and Steve Todd, December 1989. (PB90 151333/AS)
- 60 NWWS Product Retransmission Program. William R. Schneider, March 1990. (PB90 199092/AS)
- 61 A System of Collecting RAWs Data For Dissemination over AFOS. Dennis D. Gettman, January 1991. (PB91-153460)

QC
874.3
.U68
no. 62

*NOAA Western Region Computer Programs
and Problems NWS WRCP NO. 62*

CLINEWS

*Ray Stuyvesant
WSO Missoula, Montana*



May 1991

UNITED STATES
DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

National Oceanic and
Atmospheric Administration
*John A. Knauss, Under Secretary
and Administrator*

National Weather Service
*Elbert W. Friday, Jr., Assistant
Administrator for Weather Services*



*This publication has been reviewed
and is approved for publication by
Scientific Services Division,
Western Region*

Ken Mielke

*Kenneth B. Mielke, Chief
Scientific Services Division
Salt Lake City, Utah*

TABLE OF CONTENTS

	<i>PAGE</i>
<i>I. Introduction</i>	<i>1</i>
<i>II. Methodology</i>	<i>1</i>
<i>III. Software Structure</i>	<i>1</i>
<i>IV. Cautions and Restrictions</i>	<i>3</i>
<i>V. Acknowledgements</i>	<i>3</i>
<i>Part A: Program Information and Installation</i>	<i>4</i>
<i>Part B: Program Execution and Error Conditions</i>	<i>6</i>
<i>Figures</i>	
<i>CLI - output product</i>	<i>7</i>
<i>CLM - input data file</i>	<i>8</i>
<i>MCP - preformat</i>	<i>9</i>
<i>Appendix A - Program Listings</i>	<i>A.1</i>

CLINEWS

Ray Stuyvesant
WSO Missoula, Montana

I. INTRODUCTION

Daily climatological data are currently being compiled at most National Weather Service Offices. Many of these offices provide this information to the media. This program was created to minimize the effort required to compile the data. CLINEWS creates an AFOS product ready for dissemination to the media. Site-specific information can be added by editing the finished product, if needed. A by-product of this program is a set of AFOS database products (created during initial set-up using a preformat and maintained by the program) that may be used by other applications programs or as a reference for inquiries. Other than the initial creating of the preformats during set-up and the addition of any site-specific information, no manual intervention is necessary. The program can be run automatically by Watchdog.

II. METHODOLOGY

During the installation process, data files for each month of the year are created manually using a preformat. Part A provides specific instructions for entering data. Once the data files are created, CLINEWS automatically updates these files with new data. If record temperatures have changed, the user is alerted of the event.

During the execution of this program, current data are extracted from the AFOS database. Surface observations are accessed to extract the high and low temperatures and precipitation from the 6-hourly observations. The second access is to the previous CLI product for accumulated precipitation amounts. The third and final access is to the data files to extract sunrise, sunset, last year's high and low temperatures, normal and record high and low temperatures, and normal month and year-to-date precipitation.

After all the data are extracted, the CCCCLIXXX product is updated with a standard format. This product can then be used locally or disseminated as needed. (CLINEWS does NOT disseminate products.) The final task of this program is to update the data files.

III. SOFTWARE STRUCTURE

A. Description of the Program Units

1. CLINEWS

This is the main program. It extracts the local node and station name, gets the current date and time, determines which data file to extract data from, extracts the current day's data from the data file, then calls the following subroutines to manipulate the data.

2. CLID

This subroutine steps through each surface observation to find 6-hourly data for each of the last four periods. In the event of a corrected observation, only the most recent corrected report is considered. When the proper observation is found, subroutine CLIOB is called to extract the additive data. After all four observations are decoded, subroutine CLIPCP is called to compute 6, 12, and 24 hour precipitation amounts. The final step in the CLID subroutine is to convert ASCII characters to integer and correct temperatures below zero and above 99 degrees F. The range of reportable temperatures is from 40 degrees F below zero to 120 degrees F above zero.

3. CLIOB

This subroutine is a generic decoder for additive data contained in surface observations. All additive data are decoded, converted from ASCII to integer, then stored in an array called "OB". The array is returned to the calling program. RADAT data are omitted.

4. CLIPCP

This subroutine starts by retrieving the last month-to-date and year-to-date precipitation from the previous CLI message. Next, 12- and 24-hour precipitation totals are calculated. The next step is to update the month-to-date and year-to-date totals. Data are compiled and stored in an array called "PCP". This array is returned to the calling program.

5. CLICLI

CLICLI creates the finished product (CCCCLIXXX). The temperatures, precipitation, and data from the CLM files are combined into the finished product following a specific format. These data are written to a disk file. Subroutine DBSTO is called to check the data for extraneous bytes and then stores the data in the AFOS database.

6. CLICLM

This subroutine updates the monthly data file and alerts the user to new record temperatures. In addition, the current month-to-date and year-to-date precipitation totals are updated. These data should be used with caution. They are valid at 0000 UTC, no matter what time the program is run.

7. ASCINT

This subroutine converts an ASCII string into an integer. The string is checked for non-numeric characters.

8. DBSTO

This subroutine was created to check for characters that are not compatible with AFOS data processing; the major offender is the NULL character. This causes AFOS to stop its transfer of data. As a result, the end of file character is left behind. With time, this results in a number of database problems. Data are then stored in the AFOS database after being checked.

B. Running the Program

Running the program amounts to entering a command at any AFOS ADM. There are no switches and no options to be concerned with.

Simply enter, RUN:CLINEWS

The program is totally generic and extracts the node and station names from the SKEL file. Output is stored in the database and an alarm message is returned to the requesting ADM upon completion of the program.

IV. CAUTIONS AND RESTRICTIONS

CLINEWS assumes the surface observations are correct and in the proper format. The program will abort or return erroneous data if there is an error in the observation. If the CLI product is incorrect, check the observations or the CLM file. End-of-the-month totals are valid for UTC.

At WSO Missoula, CLINEWS is executed by Watchdog. The program is designed to run twice a day. The best times are just after the 0000Z and 1200Z observations.

Error checking in the program is very elaborate. However, due to the nature of the input data, not all errors can be corrected. Errors that are missed will show up as garbled or missing data in the finished product. During the test period of this program, most errors originated from the surface observations. Errors that result in the program aborting will provide messages at the ADM or dasher indicating the problem area.

CLINEWS is compatible with MAPSO-generated observations.

V. ACKNOWLEDGMENTS

This program has been produced in full cooperation with CompuSat Services.

CompuSat Services makes no claims to this product and assumes no responsibility for its content, use, or dissemination.

CLINEWS

PART A: PROGRAM INFORMATION AND INSTALLATION PROCEDURE

PROGRAM NAME: CLINEWS

AAL ID:
Revision No.: 1.0

PURPOSE: Automation of daily climatological summary and maintenance of climatological data files.

PROGRAM INFORMATION:

Development Programmer:
Ray Stuyvesant
Location: WSO Missoula
Phone: FTS 584-4840
Language: DG Fortran IV
Save File Creation Date: 2/26/91 Version 3
Running Time: 45 seconds
Disk Space: Program File, 56 RDOS Blocks

Maintenance Programmer:
Ray Stuyvesant
Location: WSO Missoula
Phone: FTS 584-4840

PROGRAM REQUIREMENTS:

<u>Program Files</u>	<u>Location</u>	<u>Comments</u>
----------------------	-----------------	-----------------

CLINEWS.SV	SYSZ	
------------	------	--

<u>Data Files</u>	<u>Location</u>	<u>Comments</u>
-------------------	-----------------	-----------------

DISKFILE	SYSZ	Deleted upon completion
DBFILE	SYSZ	Deleted upon completion

<u>AFOS Products</u>	<u>Comments</u>
----------------------	-----------------

CCCCLMJAN thru CCCCLMDEC	Input data
CCCCLIXXX	Finished product
CCCMCPCLM	Preformat for creating input data files

LOAD LINE:

RLDR/M/P CLINEWS CLID CLIOB CLIPCP CLICLI CLICLM ASCINT
DBSTO AFREAD.LB BG.LB UTIL.LB FORT.LB AFOSE.LB

PROGRAM INSTALLATION:

1. Move CLINEWS.SV from the floppy to the master directory or another directory with a link to the master directory.

2. Add the following keys to the AFOS database:

CCCCLIXXX	CCCCLMJAN	CCCCLMAPR	CCCCLMJUL	CCCCLMOCT
CCCMCPCLM	CCCCLMFEB	CCCCLMMAY	CCCCLMAUG	CCCCLMNOV
	CCCCLMMAR	CCCCLMJUN	CCCCLMSEP	CCCCLMDEC

3. Store CCCCLIXXX and CCCMCPCLM into the AFOS database using local names for CCC and XXX.
4. Enter local data for each month's CLM using the preformat (use the command M:CLM). Fill in all indicated data. Normals at the bottom of the form are optional. Include sunrise and sunset for the first of the next month at the bottom of the sunrise and sunset columns. For the column, "LAST YR TEMP", temperatures over 99 degrees F or less than -9 degrees F have to be entered by editing the CLM product after it is created using the preformat. CLM must also be edited if any precipitation entries are greater than 9.99 inches. After the CLM products are created, CLINEWS will update the fields that change. However, last year's high and low temperatures may not correspond to the days actual high and low. This is due to a lack of midnight data in AFOS. These data will need to be updated and can be done at any time.
5. Change the station name in the CCCCLIXXX to the local station name, using E:CLI at any ADM.
6. The program is ready to run. Enter: RUN:CLINEWS at any ADM.

CLINEWS

PART B: PROGRAM EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: CLINEWS

AAL ID:
Revision No.: 1.0

PROGRAM EXECUTION:

1. At any ADM, enter RUN:CLINEWS. Alarm light will indicate completion of program.

ERROR CONDITIONS:

1. ADM messages:

CLINEWS ABORTED! ERROR CONDITION: CLM FILE
Program could not access CLM data file.

CLINEWS ABORTED! ERROR CONDITION: CLI FILE
Program could not access CLI file.

CLINEWS ABORTED! ERROR CONDITION: KSRCF CLI
Program could not access CLI file.

CLINEWS ABORTED! ERROR CONDITION: CLIFILE
Input/output problem with CLI file.

CLINEWS ABORTED! ERROR CONDITION: CLMFILE
Input/cutput problem with CLM file.

DBSTO ABORTED! ERROR CONDITION: DISKFILE ERR
DBSTO subroutine could not access DISKFILE.

DBSTO ABORTED! ERROR CONDITION: DBFILE ERR
DBSTO subroutine could not access DBFILE.

DBSTO ABORTED! ERROR CONDITION: FSTORE ERR
FSTORE subroutine could not store DBFILE.

2. DASHER messages:

None

CLI - output product

TTAA00 KMSO DDHMM
DAILY CLIMATOLOGICAL SUMMARY
NATIONAL WEATHER SERVICE MISSOULA MT
601 AM MDT MON APR 9 1990

HIGH TEMPERATURE YESTERDAY	66
LOW SO FAR THIS MORNING	37
NORMAL HIGH/LOW TEMPERATURES	54 30
LAST YEAR HIGH/LOW TEMPERATURE	45 19

ALL TIME HIGH...TODAY/DATE	76	1912
ALL TIME LOW...TODAY/DATE	19	1959

SUNRISE	TODAY	7:01	TOMORROW	6:59
SUNSET	TODAY	8:16	TOMORROW	8:18

24 HOUR PRECIPITATION	0.00
TOTAL PRECIPITATION THIS MONTH	0.00
MONTHLY NORMAL THIS DATE	0.27
TOTAL PRECIPITATION THIS YEAR	1.93
ANNUAL NORMAL FOR THIS DATE	3.32

CLM - input data file

\$#TTA00 KMSO 090645

STATION NAME: WSO MISSOULA MT

APR <<MONTH	SUN		NORMAL TEMP		LAST YR TEMP		RECORD TEMPS				ACCUMULATED PRECIP			
	RISE	SET	HI	LO	HI	LO	HI	YEAR	LO	YEAR	MONTH	YEAR	MONTH	YEAR
1	7:16	8:05	51	27	70	36	78	1900	2	1936	0.03	3.08	0.00	1.93
2	7:14	8:07	51	28	75	32	75	1990	8	1936	0.06	3.11	0.00	1.93
3	7:12	8:08	51	28	68	33	74	1944	11	1935	0.09	3.14	0.00	1.93
4	7:10	8:09	52	28	65	29	75	1960	18	1920	0.12	3.17	0.00	1.93
5	7:08	8:11	52	28	57	37	78	1960	20	1936	0.15	3.20	0.00	1.93
6	7:06	8:12	53	29	65	25	78	1900	19	1928	0.18	3.23	0.00	1.93
7	7:04	8:14	53	29	66	42	77	1934	17	1973	0.21	3.26	0.00	1.93
8	7:02	8:15	54	29	66	32	82	1977	17	1973	0.24	3.29	0.00	1.93
9	7:01	8:16	54	30	45	19	76	1912	19	1959	0.27	3.32	0.00	1.93
10	6:59	8:18	55	30	55	22	77	1925	21	1933	0.30	3.35	0.00	1.93
11	6:57	8:19	55	30	60	20	77	1976	19	1975	0.33	3.38	0.00	1.93
12	6:55	8:20	55	30	65	26	80	1913	22	1895	0.36	3.41	0.00	1.93
13	6:53	8:22	56	31	69	28	79	1904	21	1970	0.39	3.44	0.00	1.93
14	6:51	8:23	56	31	70	30	80	1988	21	1928	0.42	3.47	0.00	1.93
15	6:49	8:24	57	31	62	40	83	1926	18	1973	0.45	3.50	0.00	1.93
16	6:47	8:26	57	32	57	34	81	1936	20	1896	0.48	3.53	0.00	1.93
17	6:46	8:27	57	32	50	28	86	1936	19	1896	0.51	3.56	0.00	1.9
18	6:44	8:29	58	32	68	24	88	1936	18	1977	0.54	3.59	0.00	1.93
19	6:42	8:30	58	32	72	32	83	1910	14	1951	0.57	3.62	0.00	1.93
20	6:40	8:31	58	33	80	33	84	1980	16	1927	0.61	3.66	0.00	1.93
21	6:38	8:33	59	33	72	42	82	1934	15	1951	0.65	3.70	0.00	1.93
22	6:37	8:34	59	33	55	37	83	1969	18	1951	0.69	3.74	0.00	1.93
23	6:35	8:35	59	33	55	36	84	1977	22	1967	0.73	3.78	0.00	1.93
24	6:33	8:37	60	34	53	37	87	1910	24	1967	0.77	3.82	0.00	1.93
25	6:31	8:38	60	34	56	37	90	1910	23	1907	0.81	3.86	0.00	1.93
26	6:30	8:40	60	34	57	32	82	1898	23	1962	0.85	3.90	0.00	1.93
27	6:28	8:41	61	34	47	38	85	1987	12	1907	0.89	3.94	0.00	1.93
28	6:26	8:42	61	34	53	39	87	1987	20	1907	0.93	3.98	0.00	1.93
29	6:25	8:44	61	35	61	26	86	1926	23	1950	0.97	4.02	0.00	1.93
30	6:23	8:45	62	35	69	29	82	1941	20	1909	1.01	4.06	0.00	1.93
01	6:21	8:46	=====											

NORMALS

MAX TEMP	56.5	MONTHLY HDD	633	SEASON HDD	7284
MIN TEMP	31.3	MONTHLY CDD	0	SEASON CDD	216
AVG TEMP	43.9				

MCP - preformat

TTAA00 KMSO 272316

STATION NAME:[W50 MISSOULA MT]

[MAY]<<MONTH NORMAL LAST YR

	SUN		TEMP				RECORD TEMPS				ACCUMULATED PRECIP	
	RISE	SET	HI	LO	HI	LO	HI	YEAR	LO	YEAR	NORMAL	ACTUAL
1	[5:21]	[7:46]	[62]	[35]	[48]	[27]	[82]	[1901]	[21]	[1954]	[0.04]	[4.10]
2	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
3	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
4	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
5	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
6	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
7	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
8	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
9	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
10	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
11	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
12	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
13	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
14	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
15	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
16	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
17	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
18	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
19	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
20	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
21	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
22	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
23	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
24	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
25	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
26	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
27	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
28	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
29	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
30	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
31	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
01	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

NORMALS

MAX TEMP [XX.X] MONTHLY HDD[XXXX] SEASON HDD [XXXX]
 MIN TEMP [] MONTHLY CDD [] SEASON CDD []
 AVG TEMP []

Appendix A - Program Listings

```

TYPE CLINEWS.FR
CCC=====
C PROGRAM NAME: CLINEWS
C CREATED BY: RAY STUYVESANT
C DATE: 7/1/88
C PURPOSE: MANAGES LOCAL CLIMATE DATA, CREATES CLI MESSAGE FOR MEDIA
C=====
      COMPILER NOSTACK
      INTEGER IBUF(40),KBUF(40),IDATE(3),ITIME(3),ISKEL(6),MONTH(2),
      ACLMFILE(3),PCP24,PCPMTD,PCPYTD,MX(2),MN(2),PCP(8)
      COMMON/KEY/ICLIKEY(5)/KEY2/ICLMKEY(5)/KEY3/ISAOKEY(5)
      DATA ICLIKEY/"CCCCLIXXX "/ ;
      DATA ICLMKEY/"CCCCLMXXX "/ ;
      DATA ISAOKEY/"CCCSA0XXX "/ ;
      IBYT=12 ;
      CALL DATE (IDATE,IER) ;
      CALL TIME (ITIME,IER) ;
      CALL GCHN (ICHN,IER) ;
      CALL OPENR (ICHN,"SKEL",1,IER) ;GET LOCAL NODE AND STATION ID
      CALL RDS (ICHN,ISKEL,IBYT,IER) ;
      CALL KLOSE (ICHN,IER) ;
C -----LOAD KEY DATA-----
      ICLIKEY(1)=ISKEL(5) ;LOAD LOCAL NODE AND STATION ID
      ICLIKEY(2)=IAND(177400K,ISKEL(6))+103K ;INTO KEY NAMES
      ICLIKEY(4)=ISKEL(1) ;OF DATA FILES
      ICLIKEY(5)=IAND(177400K,ISKEL(2))+60K ;
      ICLMKEY(1)=ISKEL(5) ;
      ICLMKEY(2)=IAND(177400K,ISKEL(6))+103K ;
      ISAOKEY(1)=ISKEL(5) ;
      ISAOKEY(2)=IAND(177400K,ISKEL(6))+103K ;
      ISAOKEY(4)=ISKEL(1) ;
      ISAOKEY(5)=IAND(177400K,ISKEL(2))+60K ;
C -----SET DATE & TIME-----
      I=IDATE(1) ;
      IF (ITIME(1).GE.12) GO TO 15 ;DATE/TIME GOOD
      IF (IDATE(2).NE.1) GO TO 15 ;COR DATE FOR 1ST OFMONTH
      IF (IDATE(1).NE.1) I=IDATE(1)-1 ;
      IF (IDATE(1).EQ.1) I=12 ;COR MONTH FOR 1ST OF YEAR
      IF (IDATE(1).EQ.1) IDATE(3)=IDATE(3)-1 ;COR YEAR
      IDATE(2)=31 ;ADJUSTMENTS FOR FIRST OF
      IF (I.EQ.2) IDATE(2)=28 ;MONTH AND YEAR AS NEEDED
      IF (I.EQ.4) IDATE(2)=30 ;
      IF (I.EQ.6) IDATE(2)=30 ;
      IF (I.EQ.9) IDATE(2)=30 ;
      IF (I.EQ.11) IDATE(2)=30 ;
15 GO TO (1,2,3,4,5,6,7,8,9,10,11,12),I ;DEFINE CORRECT NAME FOR
1 MONTH(1)="JA" ;CLM KEY
  MONTH(2)="N "
  GO TO 13
2 MONTH(1)="FE"
  MONTH(2)="B "
  GO TO 13
3 MONTH(1)="MA"
  MONTH(2)="R "
  GO TO 13
4 MONTH(1)="AP"
  MONTH(2)="R "
  GO TO 13
5 MONTH(1)="MA"
  MONTH(2)="Y "
  GO TO 13
C MONTH(1)=" "

```

```

0 MONTH(1)="SU"
  MONTH(2)="N "
  GO TO 13
7 MONTH(1)="JU"
  MONTH(2)="L "
  GO TO 13
8 MONTH(1)="FJ"
  MONTH(2)="G "
  GO TO 13
9 MONTH(1)="SE"
  MONTH(2)="P "
  GO TO 13
10 MONTH(1)="QC"
  MONTH(2)="T "
  GO TO 13
11 MONTH(1)="NO"
  MONTH(2)="V "
  GO TO 13
12 MONTH(1)="DE"
  MONTH(2)="C "
C -----READ LAST DATA-----
13 ICLMKEY(4)=MONTH(1) ;
  ICLMKEY(5)=MONTH(2) ;
  CALL AFREAD (1,ICLMKEY,$99) ;ACCESS CLM FILE FOR CURRENT DATA
  KREC=IDATE(2)+4 ;
  IF (ITIME(1).LT.11) KREC=KREC-1 ;
  DO 20 I=1,KREC ;GO TO CURRENT DAY
20 CALL AFREAD (2,IBUF,$99,$99) ;
  CALL AFREAD (2,KBUF,$99,$99) ;
  CALL RESET ;
C -----DECODE OBSERVATION DATA-----
  CALL CLID (IDATE,ITIME,ISAOKEY,ICLIKEY,PCP,MX,MN,IMX,IMN)
C -----CREATE NEW CLI FILE-----
  CALL CLICLI (IDATE,ITIME,ICLIKEY,IBUF,KBUF,MX,MN,PCP)
C -----UPDATE CLM DATA FILES-----
  IF (ITIME(1).GT.11) STOP
  CALL CLICLM (ICLMKEY,KREC,MN,IMN,MX,IMX,PCP,IDATE(3))
  STOP
99 CALL FORKE ("CLINEWS","CLM FILE")
  STOP
  END
H

```


TYPE CLID.FR

SUBROUTINE CLID (IDATE,ITIME,ISAOKEY,ICLIKEY,PCP,MX,MN,IMX,IMN)

```
C =====
C SUBROUTINE TO DECODE DATA FOR CLINEWS PROGRAM
C =====
  INTEGER ISAOKEY(5),ICLIKEY(5),L(128),M(256),KREC(15),ITIME(3),
  AIDATE(3),OB(10),PCP(8),P6(4),P(4),ISW6,MX(2),MNC(2),IMAX(4),IMIN(4)
  N=0
  IBLK=0
  ISW00=0
  ISW06=0
  ISW12=0
  ISW18=0

C -----
  CALL KSRCF (ISAOKEY,KREC,IER) ;OPEN SAO FOR READING
5  IF (N.EQ.4) GO TO 50 ;
  DO 10 I=1,256 ;
10  M(I)=40K ;
  CALL ROBKF (IBLK,L,IER) ;
  CALL UNPACK (L,256,M) ;
  I=48 ;
15  I=I+1 ;
  IF (M(I).NE.40K) GO TO 15 ;GO TO START OF OB
  I=I+2 ;
  IF (M(I).EQ.101K) GO TO 20 ;SA
  IF (M(I).EQ.123K) GO TO 20 ;RS
  CALL PRURF (IER) ;
  GO TO 5 ;GET NEXT OB

C -----
20  I=I+2 ;
  IF (M(I).EQ.103K) I=I+4 ;
  IF (M(I).EQ.123K) I=I+4 ;
  J=I+1 ;
  IF (M(I).EQ.61K.AND.M(J).EQ.61K) GO TO 60 ;12Z
  IF (M(I).EQ.60K.AND.M(J).EQ.65K) GO TO 70 ;06Z
  IF (M(I).EQ.62K.AND.M(J).EQ.63K) GO TO 80 ;00Z
  IF (M(I).EQ.61K.AND.M(J).EQ.67K) GO TO 90 ;18Z
  CALL PRURF (IER) ;
  GO TO 5 ;

C -----GET 6HRLY DATA-----
60  IF (ISW12.NE.0) GO TO 65 ;
  CALL CLI0B (M,OB,P(1)) ;12Z PRECIP/MN TMP
  P6(1)=OB(8) ;
  MN1=OB(1) ;
  ISW12=1 ;
  N=N+1 ;
65  CALL PRURF (IER) ;
  GO TO 5 ;
70  IF (ISW06.NE.0) GO TO 75 ;
  CALL CLI0B (M,OB,P(2)) ;06Z PRECIP/MX TMP
  P6(2)=OB(8) ;
  MX1=OB(1) ;
  ISW06=1 ;
  N=N+1 ;
75  CALL PRURF (IER) ;
  GO TO 5 ;
80  IF (ISW00.NE.0) GO TO 85 ;
  CALL CLI0B (M,OB,P(3)) ;00Z PRECIP/MX TMP
  P6(3)=OB(8) ;
  MX2=OB(1) ;
  ISW00=1 ;
  N=N+1 ;
```

```

      N=NT1
85  CALL PRURF (IER)
      GO TO 5
90  IF (ISW18.NE.0) GO TO 95
      CALL CLIOB (M.OB,P(4))
      P6(4)=OB(8)
      MN2=OB(1)
      ISW18=1
      N=N+1
95  CALL PRURF (IER)
      GO TO 5
C -----CLOSE FILES AND PROCESS PRECIP-----
50  CALL RESET
      ISW6=P(1)+P(2)+P(3)+P(4)
      CALL CLIPCP (ITIME,ICLIKEY,P6,ISW6,PCP)
C -----PROCESS TEMP-----
      IF (ITIME(1).LT.12) GO TO 100
      IMX=MX1
      IMN=MN1
      GO TO 101
100  IMX=MX2
      IMN=MN2
101  IF (IDATE(1).GE.5.AND.IDATE(1).LE.9) GO TO 110
      IF (IMN.GT.60) IMIN(2)=55K
      IF (IMN.GT.60) IMN=100-IMN
      GO TO 111
110  IF (IMX.LT.20) IMX=IMX+100
111  IMAX(1)=40K
      IMAX(2)=IMX/100
      IMAX(3)=(IMX-IMAX(2)*100)/10
      IMAX(4)=IMX-(IMAX(2)*100)-(IMAX(3)*10)+48
      IMAX(2)=IMAX(2)+48
      IMAX(3)=IMAX(3)+48
      IF (IMX.LT.100) IMAX(2)=40K
      IMIN(1)=40K
      IF (IMIN(2).NE.55K) IMIN(2)=40K
      IMIN(4)=IMN-(IMIN(3)*10)+48
      IMIN(3)=IMIN(3)+48
C -----
      CALL PACK (IMIN,4,MN)
      CALL PACK (IMAX,4,MX)
      RETURN
      END
N

```

TYPE CLICLI.FR

```
      SUBROUTINE CLICLI (IDATE,ITIME,ICLIKEY,IBUF,KBUF,MX,MN,PCP)
C =====
C SUBROUTINE TO CREATE OUTPUT FILE FOR CLINEWS PROGRAM
C =====
      INTEGER IDATE(3),ITIME(3),ICLIKEY(5),IBUF(40),KBUF(40),
      AMX(2),MN(2),PCP(8),LN(40)
      COMMON/HDG/IHDG(20)
      DATA IHDG/'CCCCLIXXX000',177777K,177777K,2400K,142600K,
      A'TTAA00 KXXX DDHMM',6412K/
      IZONE=3
      DO 10 I=1,5
10    IHDG(I)=ICLIKEY(I)
      IHDG(15)=ICLIKEY(4)
      IHDG(16)=IAND(177400K,ICLIKEY(5))+40K
      CALL DELETE ('DISKFILE',IER)
      CALL CRAND ('DISKFILE',IER)
      CALL OPENA (1,'DISKFILE',2,IER)
      CALL WRS (1,IHDG,40,IER)
      CALL AFREAD (1,ICLIKEY,$99)
      CALL AFREAD (2,LN,$99,$99)
      CALL WRS (1,LN,40,IER)
      CALL WRS (1,'<15><12>',2,IER)
      CALL AFREAD (2,LN,$99,$99)
      CALL WRS (1,LN,40,IER)
      CALL WRS (1,'<15><12>',2,IER)
      CALL MMHDR (IZONE,LN,IER)
      CALL WRS (1,LN(7),27,IER)
      CALL WRS (1,'<15><12>',2,IER)
      IF (ITIME(1).GT.11) GO TO 20
C -----WRITE THE REST OF MESSAGE-----
      CALL WRS (1,'<15><12>LOW TEMPERATURE THIS MORNING          ",38,IER)
      CALL WRS (1,MN,4,IER)
      CALL WRS (1,'<15><12>HIGH TEMPERATURE SO FAR TODAY          ",38,IER)
      CALL WRS (1,MX,4,IER)
      GO TO 21
20    CALL WRS (1,'<15><12>HIGH TEMPERATURE YESTERDAY            ",38,IER)
      CALL WRS (1,MX,4,IER)
      CALL WRS (1,'<15><12>LOW SO FAR THIS MORNING                ",38,IER)
      CALL WRS (1,MN,4,IER)
21    CALL WRS (1,'<15><12>NORMAL HIGH/LOW TEMPERATURES          ",36,IER)
      CALL WRS (1,IBUF(9),6,IER)
      CALL WRS (1,'<15><12>LAST YEAR HIGH/LOW TEMPERATURE        ",34,IER)
      CALL WRS (1,IBUF(12),8,IER)
      CALL WRS (1,'<15><12>',2,IER)
      CALL WRS (1,'<15><12>ALL TIME HIGH...TODAY/DATE            ",32,IER)
      CALL WRS (1,IBUF(17),10,IER)
      CALL WRS (1,'<15><12>ALL TIME LOW...TODAY/DATE              ",32,IER)
      CALL WRS (1,IBUF(23),10,IER)
      CALL WRS (1,'<15><12>',2,IER)
      CALL WRS (1,'<15><12>SUNRISE TODAY                          ",20,IER)
      CALL WRS (1,IBUF(3),4,IER)
      CALL WRS (1,"      TOMORROW ",14,IER)
      CALL WRS (1,KBUF(3),4,IER)
      CALL WRS (1,'<15><12>SUNSET TODAY                            ",20,IER)
      CALL WRS (1,IBUF(6),4,IER)
      CALL WRS (1,"      TOMORROW ",14,IER)
      CALL WRS (1,KBUF(6),4,IER)
      CALL WRS (1,'<15><12>',2,IER)
      CALL WRS (1,'<15><12>24 HOUR PRECIPITATION                  ", 20,IER)
```

```

CALL WRS (1, 10/12/29 HOUR PRECIPITATION      ,30, IER)
CALL WRS (1, PCP(1), 4, IER)
CALL WRS (1, "<15><12>TOTAL PRECIPITATION THIS MONTH      ", 36, IER)
CALL WRS (1, PCP(3), 6, IER)
CALL WRS (1, "<15><12>MONTHLY NORMAL THIS DATE              ", 38, IER)
CALL WRS (1, IBUF(29), 4, IER)
CALL WRS (1, "<15><12>TOTAL PRECIPITATION THIS YEAR        ", 36, IER)
CALL WRS (1, PCP(6), 6, IER)
CALL WRS (1, "<15><12>ANNUAL NORMAL FOR THIS DATE          ", 36, IER)
CALL WRS (1, IBUF(31), 6, IER)
CALL WRS (1, "<15><12>$$<203>", 5, IER)

```

C -----RESET FILES THEN STORE IN DATABASE-----

```

CALL RESET
CALL DBSTO (IER)
IF (IER.NE.1) GO TO 99
RETURN
99 CALL FORKE ("CLINEWS", "CLIFILE")
CALL RESET
STOP
END

```

H

TYPE CLICLM.FR

SUBROUTINE CLICLM (ICLMKEY,KREC,MN,IMN,MX,IMX,PCP,IDATE)

C=====

C SUBROUTINE TO UPDATE CLM FILE FOR CLINEWS PROGRAM

C=====

INTEGER ICLMKEY(5),IBUF(40),MN(2),MX(2),UDATE(4),DATE(2),JH(40),
APCP(8)

COMMON/HDG2/JHDG(20)

DATA JHDG/"CCCCLMXXX000", 17777K, 17777K, 2400K, 142600K,
A"TTA000 KXXX DDHMM",6412K/

C -----LOAD HEADER AND DATE/TIME-----

DO 5 I=1,4 ;LOAD CLM FILE NAME
5 JHDG(I)=ICLMKEY(I) ;
JHDG(5)=IAND(177400K,ICLMKEY(5))+60K ;
UDATE(1)=IDATE/1000 ;COMPUTE 4-DIGIT DATE FOR
UDATE(2)=(IDATE-(UDATE(1)*1000))/100 ;RECORD TEMPS
UDATE(3)=(IDATE-(UDATE(1)*1000)-(UDATE(2)*100))/10
UDATE(4)=IDATE-(UDATE(1)*1000)-(UDATE(2)*100)-(UDATE(3)*10)+48
UDATE(3)=UDATE(3)+48 ;
UDATE(2)=UDATE(2)+48 ;
UDATE(1)=UDATE(1)+48 ;
CALL PACK (UDATE,4,DATE) ;PACK NEW DATE
CALL DELETE ("DISKFILE",IER) ;DELETE OLD DISKFLIE
CALL CRAND ("DISKFILE",IER) ;CREATE NEW DISKFILE
CALL OPENA (1,"DISKFILE",2,IER1)

C -----LOAD DATA-----

CALL WRS (1,JHDG,40,IER2) ;WRITE HEADING TO DISKFILE
CALL AFREAD (1,ICLMKEY,\$99) ;READ CLM FILE ONE LINE AT A
DO 10 I=1,40 ;TIME
CALL AFREAD (2,IBUF,\$99,\$99) ;READ NEXT LINE
IF (I.NE.KREC) GO TO 15 ;WRITE LAST LINE UNLESS IT
IBUF(12)=MX(1) ;IS TODAY'S LINE
IBUF(13)=MX(2) ;IF TODAY'S LINE, UPDATE
IBUF(14)=MN(1) ;DATA THEN WRITE TO DISKFILE
IBUF(15)=MN(2) ;
CALL ASCINT (IBUF,33,4,JMX) ;RECORD HI
CALL ASCINT (IBUF,45,4,JMN) ;RECORD LO
IF (JMX.GT.IMX) GO TO 20 ;NO RECORD GO TO 20
IBUF(17)=MX(1) ;ELSE WRITE NEW RECORD
IBUF(18)=MX(2) ;
IBUF(20)=DATE(1) ;
IBUF(21)=DATE(2) ;
CALL FORKO ("CLINEWS","RECORD HI",IER)
20 IF (IMN.GT.JMN) GO TO 25 ;NO RECORD GO TO 15
IBUF(23)=MN(1) ;ELSE WRITE NEW RECORD
IBUF(24)=MN(2) ;
IBUF(26)=DATE(1) ;
IBUF(27)=DATE(2) ;
CALL FORKO ("CLINEWS","RECORD LO",IER)

C -----DO PRECIP-----

25 IBUF(34)=PCP(3) ; INSERT PCP MTD
IBUF(35)=PCP(4) ;
IBUF(36)=PCP(5) ;
IBUF(37)=PCP(6) ; INSERT PCP YTD
IBUF(38)=PCP(7) ;
IBUF(39)=PCP(8) ;
15 CALL WRS (1,IBUF,80,IER) ; WRITE IT ALL TO DISKFILE
10 CONTINUE ;

C -----DO EOF RESET FILES AND STORE IN DATABASE-----

CALL WRS (1,"<203>",1,IER) ;WRITE END OF FILE
CALL RESET ;
CALL DBSTO (IER) ;STORE IN DATA BASE
IF (IER.NE.1) GO TO 99
RETURN
99 CALL FORKE ("CLINEWS","CLMFILE")
CALL RESET
RETURN
END

H

!!B WD RUN: ROBTIME

!!B WD RUN COMPLETE

H

TYPE CLI0B.FR

SUBROUTINE CLI0B (M,OB,ISW6)

```
C =====
C SUBROUTINE TO EXTRACT ADDITIVE DATA FROM SURFACE OBSERVATION
C =====
      INTEGER M(256),OB(10),ISW6
      ISW6=0
      J=52
      DO 5 I=1,10 ;CLEAR OB VECTOR
5     OB(I)=0 ;
      DO 10 I=53,256 ;SMOOTH OB
      IF (M(I).EQ.12K) GOTO 10 ;DELETE LF
      IF (M(I).EQ.15K) M(I)=40K ;REPLACE CR WITH SP
      IF (M(I-1).EQ.40K.AND.M(I).EQ.40K) GO TO 10 ;SKIP DBL SP
      J=J+1 ;INCREMENT NEW OB
      M(J)=M(I) ;INCLUDE CHARACTER
      IF (M(J).EQ.203K) GOTO 20 ;EOF
10    CONTINUE ;
      RETURN ;NO EOF
C -----FIND TEMP-----
20    IF (M(J).NE.40K) GOTO 22 ;SPACE
      IF (M(J-3).EQ.40K) GOTO 24 ;SPACE
22    J=J-1 ;
      GO TO 20 ;
C -----DECODE TEMP-----
24    OB(1)=(M(J-2)-48)*10+M(J-1)-48 ;
C -----DECODE 2 GROUP-----
      IF (M(J+1).NE.62K) GO TO 30 ;2
      OB(2)=(M(J+2)-48)*1000+(M(J+3)-48)*100 ;
      OB(2)=OB(2)+(M(J+4)-48)*10+M(J+5)-48 ;
C -----DECODE 904 & 902 GROUPS-----
30    J=J-3 ;
      IF (M(J-5).NE.71K) GOTO 60 ;9
      IF (M(J-3).EQ.64K) OB(3)=(M(J-2)-48)*10+M(J-1)-48 ;4
      IF (M(J-3).EQ.62K) OB(4)=(M(J-2)-48)*10+M(J-1)-48 ;2
      J=J-6 ;
      IF (M(J-5).NE.71K) GOTO 60 ;9
      IF (M(J-3).EQ.64K) OB(3)=OB(3)+100 ;904 +100
      IF (M(J-3).EQ.62K) OB(4)=(M(J-2)-48)*10+M(J-1)-48 ;2
      J=J-6 ;
      IF (M(J-5).NE.71K) GOTO 60 ;9
      OB(4)=(M(J-2)-48)*10+M(J-1)-48 ;2
      J=J-6 ;
C -----DECODE CLOUD GROUP-----
C
C
C
60    IF (M(J-5).NE.40K) GOTO 70 ;SPACE
      OB(5)=M(J-3)-48 ;LOW CLOUD
      OB(6)=M(J-2)-48 ;MID CLOUD
      OB(7)=M(J-1)-48 ;HIGH CLOUD
      J=J-5 ;
C -----DECODE PRECIP AND APP-----
70    IF (M(J-4).EQ.40K) GOTO 80 ;
      OB(8)=(M(J-2)-48)*10+M(J-1)-48 ;PRECIP
      OB(9)=(M(J-4)-48)*10+M(J-3)-48 ;PRES CHG
      OB(10)=M(J-5)-48 ;PRES TEND
      IF (OB(8).EQ.0) ISW6=1 ;SET FOR TRACE
      TYPE (OB(I),I=1,10) ;
      RETURN ;
C -----DECODE APP ONLY-----
80    OB(9)=(M(J-2)-48)*10+M(J-1)-48 ;PRES CHG
      OB(10)=M(J-3)-48 ;PRES TEND
C -----
      TYPE (OB(I),I=1,10)
      RETURN
      END
```

H

TYPE CLIPCP.FR

SUBROUTINE CLIPCP (ITIME,ICLIKEY,P6,ISW6,PCP)

```
C =====
C PROGRAM TO COMPUTE PRECIP DATA FOR CLINEWS PROGRAM
C =====
      INTEGER ITIME(3),ICLIKEY(5),IBUF(128),PCPMTD(6),PCPYTD(6),ISW6,
      APCPM,PCPY,PCP12,PCP24,P6(4),PM(6),PY(6),P(4),PCP(8)
      ISWM=0
      ISWY=0
      CALL AFREAD (1,ICLIKEY,$98)           ;GET CURRENT TOTALS
5     CALL AFREAD (2,IBUF,$99,$99)         ;READ MTD
      IF (IBUF(1).NE."TO") GO TO 5
      IF (IBUF(20).EQ."TR") ISWM=1         ;SET MTD TR SWITCH
      CALL UNPACK (IBUF(18),6,PCPMTD)      ;UNPACK MTD
      CALL AFREAD (2,IBUF,$99,$99)         ;SKIP
      CALL AFREAD (2,IBUF,$99,$99)         ;READ YTD
      IF (IBUF(20).EQ."TR") ISWY=1         ;SET YTD TR SWITCH
      CALL UNPACK (IBUF(18),6,PCPYTD)      ;UNPACK YTD
      DO 10 I=2,6
      IF (PCPMTD(I).LT.48) PCPMTD(I)=48     ;SET ALL TO NUMBERS
      IF (PCPYTD(I).LT.48) PCPYTD(I)=48     ; ONLY
      IF (PCPMTD(I).GT.57) PCPMTD(I)=48
      IF (PCPYTD(I).GT.57) PCPYTD(I)=48
10    CONTINUE
C -----24 HOUR PRECIP-----
      IF (ITIME(1).LT.12) PCP12=P6(3)+P6(4) ;PM PRECIP
      IF (ITIME(1).GE.12) PCP12=P6(1)+P6(2) ;AM PRECIP
      PCP24=P6(1)+P6(2)+P6(3)+P6(4)        ;24 HR PRECIP
      IF (PCP24.GT.0) GO TO 35
      IF (ISW6.NE.0) PCP(2)="TR"           ;SWITCH FOR TRACE
      IF (ISW6.NE.0) GO TO 40
35    P(1)=PCP24/100
      P(2)=46
      P(3)=(PCP24-P(1)*100)/10
      P(4)=(PCP24-P(1)*100)-(P(3)*10)+48
      P(1)=P(1)+48
      P(3)=P(3)+48
      CALL PACK (P,4,PCP)                  ;PUT 24 HOUR PRECIP INTO PCP ARRAY
C -----MONTH TO DATE PRECIP-----
40    IF (PCP12.GT.0) GO TO 45
      IF (ISWM.NE.0) PCP(3)="TR"           ;MTD A TRACE
      IF (ISWM.NE.0) GO TO 50
45    PCPM=((PCPMTD(2)-48)*1000)+((PCPMTD(3)-48)*100)
      PCPM=PCPM+((PCPMTD(5)-48)*10)+(PCPMTD(6)-48)+PCP12
      PM(1)=32
      PM(2)=PCPM/1000
      PM(3)=(PCPM-(PM(2)*1000))/100
      PM(4)=46
      PM(5)=(PCPM-(PM(2)*1000)-(PM(3)*100))/10
      PM(6)=PCPM-(PM(2)*1000)-(PM(3)*100)-(PM(5)*10)+48
      PM(2)=PM(2)+48
      PM(3)=PM(3)+48
      PM(5)=PM(5)+48
      IF (PM(2).EQ.48) PM(2)=40K
      CALL PACK (PM,6,PCP(3))             ;PUT MTD PRECIP INTO PCP ARRAY
C -----YEAR TO DATE PRECIP-----
50    IF (PCP12.GT.0) GO TO 55
      IF (ISWY.NE.0) PCP(8)="TR"           ;YTD A TRACE
      IF (ISWY.NE.0) GO TO 97
55    PCPY=((PCPYTD(2)-48)*1000)+((PCPYTD(3)-48)*100)
      PCPY=PCPY+((PCPYTD(5)-48)*10)+(PCPYTD(6)-48)+PCP12
```

```

PCPY=PCPY*(PCPY/1000)/740/410/PCPY/1000/740/PCPY/1000
PY(1)=40K ;CONVERT YTD PRECIP
PY(2)=PCPY/1000 ; TO ASCII
PY(3)=(PCPY-(PY(2)*1000))/100 ;
PY(4)=46 ;
PY(5)=(PCPY-(PY(2)*1000)-(PY(3)*100))/10 ;
PY(6)=PCPY-(PY(2)*1000)-(PY(3)*100)-(PY(5)*10)+48 ;
PY(2)=PY(2)+48 ;
PY(3)=PY(3)+48 ;
PY(5)=PY(5)+48 ;
IF (PY(2).EQ.48) PY(2)=40K
CALL PACK (PY,6,PCP(6))

```

```

C -----RESET FILES AND RETURN-----
97 CALL RESET
RETURN
98 CALL FORKE ("CLINEWS","KSRCF CLI")
CALL RESET
RETURN
99 CALL FORKE ("CLINEWS","CLI FILE")
CALL RESET
RETURN
END
H

```

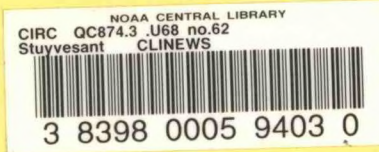

TYPE DBSTO.FR

```
=====
C PROGRAM NAME: DBSTO
C PROGRAMMER: RAY STUYVESANT
C DATE: 3/7/89
C PURPOSE: CHECKS FORMAT OF DISK FILE BEFORE IT IS CALLED
=====
C
SUBROUTINE DBSTO (IER)
INTEGER IBUF(40),JBUF(80)
IBYT=40
CALL GCHN (ICHN,IER)
CALL OPENR (ICHN,"DISKFILE",1,IER) ;OPEN CHANNEL TO DISK FILE
IF (IRE.NE.0) GO TO 95 ;CHECK FOR ERROR
CALL DELETE ("DBFILE",IER) ;DELETE OLD SCRATCH FILE
CALL CRAND ("DBFILE",IER) ;CREATE NEW SCRATCH FILE
CALL GCHN (JCHN,IER) ;
CALL OPENA (JCHN,"DBFILE",1,IER) ;OPEN CHANNEL TO NEW FILE
IF (IER.NE.0) GO TO 96 ;CHECK FOR ERROR
CALL RDS (ICHN,IBUF,IBYT,IER) ;READ FIRST LINE CCCNNNXXX
CALL WRS (JCHN,IBUF,IBYT,IER) ;WRITE FIRST LINE TO NEW FILE
CALL RDS (ICHN,IBUF,IBYT,IER) ;READ SECOND LINE TTA000
CALL WRS (JCHN,IBUF,IBYT,IER) ;WRITE SECOND LINE TO NEW FILE
C
DO 10 I=1,240 ;CHECK FILE 1 LINE AT A TIME
CALL RDS (ICHN,IBUF,IBYT,IER) ;READ NEXT LINE
CALL UNPACK (IBUF,80,JBUF) ;UNPACK LINE
DO 20 J=1,IBYT ;CHECK BYTE BY BYTE FOR:
IF (JBUF(J).EQ.203K) GO TO 50 ; END OF FILE
IF (JBUF(J).EQ.12K) GO TO 20 ; CR
IF (JBUF(J).EQ.15K) GO TO 20 ; LF
IF (JBUF(J).LE.37K) JBUF(J)=40K ; NULLS, CTRL CHARACTERS
IF (JBUF(J).GE.176K) JBUF(J)=40K ; EXTENDED CHARACTERS
20 CONTINUE ;
CALL PACK (JBUF,80,IBUF) ;PACK CURRENT LINE
CALL WRS (JCHN,IBUF,IBYT,IER) ;WRITE CURRENT LINE TO NEW FILE
10 CONTINUE ;
CALL FORKE ("DBSTO","NO EOF") ;NO END OF FILE
STOP ;
50 CALL PACK (JBUF,80,IBUF) ;PACK LAST LINE
CALL WRS (JCHN,IBUF,IBYT,IER) ;WRITE LAST LINE PLUS EOF
CALL KLOSE (ICHN,IER) ;CLOSE CHANNELS
CALL KLOSE (JCHN,IER) ;
CALL FSTORE ("DBFILE",0,IER) ;STORE DISKFILE INTO DATABASE
IF (IER.NE.0) GO TO 97 ;
RETURN ;RETURN TO CALLING PROGRAM
C
95 CALL FORKE ("DBSTO","DISKFILE ERR")
RETURN
96 CALL FORKE ("DBSTO","DBFILE ERR")
RETURN
97 CALL FORKE ("DBSTO","FSTORE ERR")
RETURN
END
W
```

```

TYPE ASCINT.FR
SUBROUTINE ASCINT (M,JBGN,JEND,L)
C M=PACK ARRAY JBGN=1ST CHARACTER JEND=# OF CHARACTERS L=INTEGER
C SUBROUTINE TO CONVERT ASCII CHARACTERS TO INTEGER VALUES
INTEGER M(40),M1(80)
LOGICAL NEG
L=0
NEG=.FALSE.
IBGN=JBGN
IEND=JBGN+JEND ;INCLUDE SPACE AFTER LAST DIGIT
CALL UNPACK (M,80,M1)
DO 5 I=IBGN,IEND ;SCREEN FOR INTEGER ONLY
IF (M1(I).EQ.45) NEG=.TRUE. ;CHECK IF LESS THAN ZERO
IF (M1(I).LT.48.OR.M1(I).GT.57) M1(I)=32 ;USE NUMBERS ONLY
5 CONTINUE
J=IBGN
K=IEND
DO 10 I=IBGN,IEND ;FIND FIRST DIGIT
IF (M1(I).EQ.32) J=J+1
IF (M1(I).NE.32) GO TO 11 ;LAST DIGIT HAS BEEN READ
IF (J.EQ.IEND) RETURN ;NO VALID INTEGER
10 CONTINUE
DO 15 I=J,K ;FIND LAST DIGIT
IF (M1(I).EQ.32) GO TO 16
15 CONTINUE
K=I-1 ;MAKE K LAST DIGIT
DO 20 I=J,K
L=L*10+M1(I)-48 ;COMPUTE INTEGER VALUE
20 CONTINUE
IF (NEG) L=-L ;MAKE NEGATIVE IF TRUE
RETURN
END
N

```



NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

The National Oceanic and Atmospheric Administration was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to assess the socioeconomic impact of natural and technological changes in the environment and to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth.

The major components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications.

PROFESSIONAL PAPERS--Important definitive research results, major techniques, and special investigations.

CONTRACT AND GRANT REPORTS--Reports prepared by contractors or grantees under NOAA sponsorship.

ATLAS--Presentation of analyzed data generally in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

TECHNICAL SERVICE PUBLICATIONS--Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

TECHNICAL REPORTS--Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS--Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.



Information on availability of NOAA publications can be obtained from:

NATIONAL TECHNICAL INFORMATION SERVICE

U. S. DEPARTMENT OF COMMERCE

5285 PORT ROYAL ROAD

SPRINGFIELD, VA 22161

