

**PART 2: PROCESSING CIMIS METEOROLOGICAL FILES
TO PRODUCE ISCST3-READY WEATHER FILES:
TECHNICAL MANUAL**

**Bruce Johnson
Research Scientist III**

**Edgar Vidrio
Environmental Scientist**

July 2011
Rev Jan 2014

California Environmental Protection Agency
Department of Pesticide Regulation
Environmental Monitoring Branch
Air Program
P.O. Box 4015
Sacramento, California 95812-4015

Table of Contents

1. Introduction	3
2. Objectives	4
3. Description of the programs	4
3.1. UNIXorDOSV2.FOR	4
<i>Command: UNIXorDOSV2 CSS-YY.UXT (User Manual Pg 19)</i>	4
3.2. UNIX2DOS.EXE	5
<i>Command: UNIX2DOS CSS-YY.UXT CSS-YY.DXT (User Manual pg19)</i>	5
3.3. FIXDASHV2.FOR	6
<i>Command: FIXHDASHV2 CSS-YY.DXT CSS-YY.TXT (User Manual pg19)</i>	6
3.4. PREFORMV2.FOR	7
<i>Command: PREFORMV2 CSS-YY.TXT CSS-YY.Pre (User Manual pg20)</i>	8
3.5. CIMPRO1V4.FOR.....	8
<i>Command: CIMPRO1V4 CSS-YY.PRE CSS-YY.DET (User Manual pg20)</i>	8
3.6. CIMPRO2V5.FOR.....	10
<i>Command: CIMPRO2V5 CSS-YY.DET YYCIMSSS.2PC (User Manual pg21)</i>	10
3.7. MAKEISCV2.FOR.....	18
<i>Command: MAKISCV2 YYCIMSSS.2PC YYCIMSSS.ISC (User Manual pg22)</i>	18
3.8. Program Listings.....	18
4. Summary	18
5. References	19
APPENDIX 1: PROGRAM LISTINGS	20
APPENDIX 2: Manual versus CIMPRO2V5 determined stability classifications	77

1. Introduction

This technical manual is a companion to Part 1 (Vidrio and Johnson 2011), which describes a database of processed California Irrigation Management Information System (CIMIS) meteorological data. The programs used to process that data are described in more detail in this document.

The programs, CIMPRO1V3.FOR and CIMPRO2V5.FOR, were based on two programs originally created by Craig Schmidt of DowElanco (now Dow AgroSciences) in the early 1990s. His programs, in turn, were based on the USEPA PCRAMMET meteorological pre-processor. He modified the PCRAMMET program to utilize standard deviation of lateral wind direction and night/day to determine stability class (USEPA 2000).

Since then, we have modified his original programs for various uses. Probably the largest previous effort was in developing the meteorological data set for investigating methyl bromide buffer zones (Johnson 2001). More recent efforts include developing meteorological data for investigating public health issues in Kettleman City, CA.

The programs have been primarily for internal use by one of us (Johnson). As such, they were quirky and obscure. Along with creating the CIMIS database of processed data in Part 1 (Vidrio and Johnson 2011), we decided to simplify running the programs by creating a standard syntax. There are six FORTRAN programs and one Turbo-C program (UNIX2DOS). Five of the six FORTRAN processing programs (UNIXORDOSV2 is the exception) and the Turbo-C program all utilize the following syntax within a DOS command window as follows:

```
program inputfile outputfile
```

Where program is the name of the executable program, and inputfile and outputfile are the named input and output files. The input file must exist. The outputfile may or may not exist. If it does exist, it is overwritten. Part 1 (Vidrio and Johnson 2011) provide more detailed instructions on using the programs. Two programs were more recently written. The FORTRAN program, UNIXORDOSV2, was recently added in order assess the line terminator style in downloaded CIMIS data. FIXDASHV2 was added to automate the substitution of missing value codes for double dashes which appear in CIMIS data files.

Except for UNIX2DOS, the other six programs were written and compiled in Lahey FORTRAN 95. The following option switches controlled the compilation:

```
: -nap -nc -chk -co -ncover -dal -ndbl -ndll -nf95 -fix  
: -ng -nin -info -lst -nlong -maxfatals 50 -nml -o0 -no  
: -npause -nprivate -npca -nquad -nsav -nstaticlink -stchk  
: -nswm -tp -trace -ntrap -nvsw -w -winconsole -nwisk -nwo  
: -xref -zero
```

This technical document is written with the assumption that the user is following the file naming conventions located in Part 1 of this report (Vidrio and Johnson 2011). The user should be aware that these programs are provided 'as is', with no warranty, implied or expressed. Also, the resulting data files may have missing days and/or estimated values and may include data that CIMIS has flagged for various reasons. Based on reviewer suggestion, the generic command for each program unit is included at the beginning of each program section with a reference to the appropriate page in the User Manual.

2. Objectives

The objective of this document is to provide the technical details of the programs used to process the CIMIS data and provide listings of the current versions.

3. Description of the programs

3.1. UNIXorDOSV2.FOR

Command: UNIXorDOSV2 CSS-YY.UXT (User Manual Pg 19)

This utility is the only utility that does not have a user-specified output file. UNIXorDOS produces a temporary file called UORD.TMP and its purpose is for debugging. It provides additional information about the file CSS-YY.UXT and is not part of the CIMIS met processing sequence. UORD.TMP can be safely deleted if desired.

The purpose of UNIXorDOSV2 is to determine if the UXT file is Unix style or DOS style with regard to the line terminators. Files with Unix-style line terminators cannot be processed by the FORTRAN programs described in this document. At this time, we are unable to determine if Department of Water Resources downloaded CIMIS data will always be in the Unix style or DOS style. The style changed from Unix to DOS sometime in the first half of 2011, but as far as we can determine, there is no policy regarding this issue.

If a DOS style file is mistakenly processed with UNIX2DOS, it will corrupt the file and make the file unusable for subsequent processing. Therefore, it is important to determine whether the file downloaded from CIMIS is in Unix or DOS style. The next section goes into greater explanation of what Unix or DOS style means. Briefly, the line terminators for Unix are 0A hexadecimal (10 decimal, also called line feed = LF, or ^J) and for DOS the line terminator consists of two characters 0D0A hexadecimal (0D is 13 decimal, also called carriage return = CR, or ^M). The line terminator tells the processing programs where each record (=line) ends. UNIXorDOSV2 examines the first 10,000 bytes of the input file and looks for the occurrence of 0A or 0D0A in order to determine if the file is a Unix style or DOS style file.

As the tutorial in the user manual explains, after changing the filename to CSS-YY.UXT, you need to run UNIXorDOS to determine if you must run the UNIX2DOS utility. If you

do not need to run UNIX2DOS, then rename the CSS-YY.UXT file into CSS-YY.TXT and skip the next step. If you do not need to run UNIX2DOS, then skip to step 3.3.

3.2. UNIX2DOS.EXE

Command: UNIX2DOS CSS-YY.UXT CSS-YY.DXT (User Manual pg19)

This utility was written in Turbo C by unknown authors. It converts a Unix-based line feed (hexadecimal 0A) text file to a DOS based carriage return (hexadecimal 0D), line feed for the line terminator convention. It inserts a carriage return (hexadecimal 0A) before each line feed found in the Unix text file. *The UNIX2DOS program should only be used if UNIXorDOSV2 program indicates that UNIX2DOS needs to be run.* Figure 3.2.1 shows the Unix text style, which does not use the carriage return Above the line are the ascii characters. Below the line on the left are the hexadecimal characters. Each line ends with 0A, which is a line feed. This is represented on the lower right as a "J". After processing with UNIX2DOS, the same extract shows the insertion of the carriage return with the 0D inserted just before the 0A (represented by M in lower right Figure 3.2.2) before each line feed (represented by J lower right Figure 3.2.2).

```

0092,01/01/2005,0100,001,*, -6,N,--,*, 49.0,*, 2.4,*, 189.2,*, 46.8,*
0092,01/01/2005,0200,001,*, -6,N,--,*, 48.3,*, 2.2,*, 145.4,*, 53.6,*
0092,01/01/2005,0300,001,*, -6,N,--,*, 48.0,*, 3.3,*, 125.3,*, 13.8,*
0092,01/01/2005,0400,001,*, -6,N,--,*, 47.6,*, 4.5,*, 142.5,*, 11.9,*
0092,01/01/2005,0500,001,*, -6,N,--,*, 47.7,*, 4.7,*, 149.7,*, 14.6,*
0092,01/01/2005,0600,001,*, -5,N,--,*, 47.2,*, 4.5,*, 148.6,*, 12.9,*
0092,01/01/2005,0700,001,*, -4,N,--,*, 46.2,*, 2.0,*, 130.6,*, 29.4,*
P:\CIMIS\TECH-M\c92-05.UXT File 1 Pg 1 Ln 1 Pos 1
30 30 39 32 2C 30 31 2F 30 31 2F 32 30 30 35 2C 0092,01/01/2005,
30 31 30 30 2C 30 30 31 2C 2A 2C 20 20 20 2D 36 0100,001,*, -6
2C 4E 2C 2D 2D 2C 2A 2C 20 20 20 34 39 2E 30 2C ,N,--,*, 49.0,
2A 2C 20 20 20 20 32 2E 34 2C 2A 2C 20 20 31 38 *, 2.4,*, 18
39 2E 32 2C 2A 2C 20 20 20 34 36 2E 38 2C 2A 2C 9.2,*, 46.8,*,
20 20 20 20 30 2E 30 30 0A 0.00J
30 30 39 32 2C 30 31 2F 30 31 2F 32 30 30 35 2C 0092,01/01/2005,
30 32 30 30 2C 30 30 31 2C 2A 2C 20 20 20 2D 36 0200,001,*, -6
2C 4E 2C 2D 2D 2C 2A 2C 20 20 20 34 38 2E 33 2C ,N,--,*, 48.3,
2A 2C 20 20 20 20 32 2E 32 2C 2A 2C 20 20 31 34 *, 2.2,*, 14
35 2E 34 2C 2A 2C 20 20 20 35 33 2E 36 2C 2A 2C 5.4,*, 53.6,*,
20 20 20 20 30 2E 30 30 0A 0.00J
30 30 39 32 2C 30 31 2F 30 31 2F 32 30 30 35 2C 0092,01/01/2005,
30 33 30 30 2C 30 30 31 2C 2A 2C 20 20 20 2D 36 0300,001,*, -6
2C 4E 2C 2D 2D 2C 2A 2C 20 20 20 34 38 2E 30 2C ,N,--,*, 48.0,

```

Figure 3.2.1 Raw download from Unix servers on CIMIS website showing Unix line termination. Two lines from station 92 met data shown with characters in hexadecimal format. Each line is terminated with hex 0A, which is called line feed in the ASCII table and represented by J in the characters to the right of the hexadecimal listing.

```

c:\ cmd - pe c92-05.dxt
0092,01/01/2005,0100,001,*, -6,N, -99,*, 49.0,*, 2.4,*, 189.2,*, 46.
0092,01/01/2005,0200,001,*, -6,N, -99,*, 48.3,*, 2.2,*, 145.4,*, 53.
0092,01/01/2005,0300,001,*, -6,N, -99,*, 48.0,*, 3.3,*, 125.3,*, 13.
0092,01/01/2005,0400,001,*, -6,N, -99,*, 47.6,*, 4.5,*, 142.5,*, 11.
0092,01/01/2005,0500,001,*, -6,N, -99,*, 47.7,*, 4.7,*, 149.7,*, 14.
0092,01/01/2005,0600,001,*, -5,N, -99,*, 47.2,*, 4.5,*, 148.6,*, 12.
0092,01/01/2005,0700,001,*, -4,N, -99,*, 46.2,*, 2.0,*, 130.6,*, 29.
P:\CIMIS\TECH-M\1\C92-05.DXT File 1 Pg 1 Ln 1 Pos 1
30 30 39 32 2C 30 31 2F 30 31 2F 32 30 30 35 2C 0092,01/01/2005,
30 31 30 30 2C 30 30 31 2C 2A 2C 20 20 20 2D 36 0100,001,*, -6
2C 4E 2C 20 20 2D 39 39 2C 2A 2C 20 20 20 34 39 ,N, -99,*, 49
2E 30 2C 2A 2C 20 20 20 20 32 2E 34 2C 2A 2C 20 .0,*, 2.4,*,
20 31 38 39 2E 32 2C 2A 2C 20 20 34 36 2E 38 189.2,*, 46.8
2C 2A 2C 20 20 20 20 30 2E 30 30 0D 0A ,*, 0.00MJ
30 30 39 32 2C 30 31 2F 30 31 2F 32 30 30 35 2C 0092,01/01/2005,
30 32 30 30 2C 30 30 31 2C 2A 2C 20 20 20 2D 36 0200,001,*, -6
2C 4E 2C 20 20 2D 39 39 2C 2A 2C 20 20 20 34 38 ,N, -99,*, 48
2E 33 2C 2A 2C 20 20 20 20 32 2E 32 2C 2A 2C 20 .3,*, 2.2,*,
20 31 34 35 2E 34 2C 2A 2C 20 20 35 33 2E 36 145.4,*, 53.6
2C 2A 2C 20 20 20 30 2E 30 30 0D 0A ,*, 0.00MJ
30 30 39 32 2C 30 31 2F 30 31 2F 32 30 30 35 2C 0092,01/01/2005,
30 33 30 30 2C 30 30 31 2C 2A 2C 20 20 20 2D 36 0300,001,*, -6
2C 4E 2C 20 20 2D 39 39 2C 2A 2C 20 20 20 34 38 ,N, -99,*, 48

```

Figure 3.2.2 After running the UNIX2DOS program on a Unix formatted file. OD (hex) has been inserted before each 0A (hex) for line termination.

Using a program like Notepad to examine the raw data file from Unix results in a difficult-to-read screen of text with no breaks between lines as the text wraps around. After insertion of the carriage returns, the text appears as a flat, fixed column set of records, which line up vertically. The carriage return insertion readies the file for further processing.

3.3. FIXDASHV2.FOR

Command: FIXHDASHV2 CSS-YY.DXT CSS-YY.TXT (User Manual pg19)

The CIMIS output files contain missing values. These are indicated by the use of a double dash with the field collapsed. Previously these missing value situations were edited by hand, using NOTEPAD or something equivalent to replace the dashes with a missing value code of -99, -99.0 or -99.00 and an appropriate number of spaces in order to line up the fields and commas (Figure 3.3). In order to prepare the file for use with PREFORMV2, the next step in processing, the fields must line up. FIXDASHV2 automates this replacement process. This program basically reads the input file, line by line, and determines field by field if there is a missing value. If so, the missing value is replaced by the appropriate form of -99 (the two radiation fields use -99, while the other fields use a decimal place). Also, there are instances where the value is missing in the downloaded CIMIS file as indicated by the double dashes, but the QC flag is an 'E', 'Q', 'S' or 'T'. FIXDASHV2 changes the 'E' to an 'M'. The CIMIS web page states for the E and T flags: "Historical average of one of the sensors used to calculate a parameter." and for the 'S' flag: "Sensor is not in service or data is out of sensor threshold". And the CIMIS website describes the 'Q' flag as "All quality control could not be performed because a comparison sensor is severe". However, the "M" QC flag defines a field as missing, which is the case when two dashes are present.

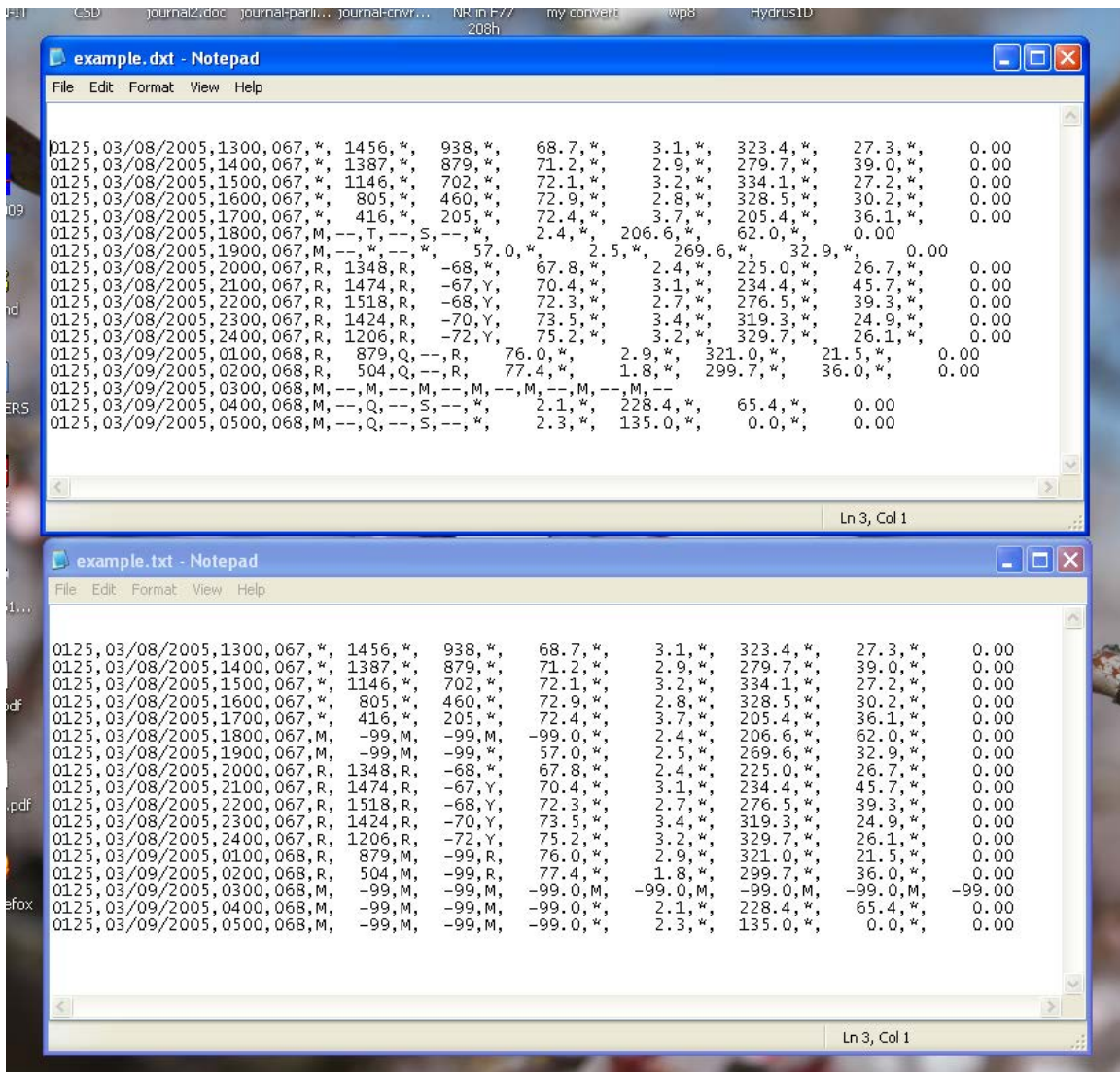


Figure 3.3 After running FIXDASHV2 on the top portion, the bottom portion results. The -99 code has been inserted for the dashed fields and the missing value code 'M' has been inserted in those cases. The fields now line up.

FIXDASHV2 was tested on 95 files which contained missing values. Surprisingly, one file contained '#####' instead of '-' for the missing values (Station 125, Arvin Edison, March 4 and 5 in 2005). As a result, FIXDASHV2 was modified to look for ##### in the solar radiation field as a special case. It is conceivable that other CIMIS files will contain unanticipated forms which may require manual editing and formatting.

FIXDASHV2 counts the number of commas in each line. There should be 17 commas. If not, then FIXDASHV2 terminates with an error message and prints out the suspect line. If this situation arises, that line will have to be edited manually.

3.4. PREFORMV2.FOR

Command: PREFORMV2 CSS-YY.TXT CSS-YY.Pre (User Manual pg20)

In the mid to late 1990s, the CIMIS program changed the format of the downloaded output files. PREFORMV2.FOR was written to reformat the revised format to match the old format, as this was easier than rewriting the processing programs. In addition, it performs some rudimentary checking of the data files. Specifically, it checks the position of the commas, slashes (used in dates) and decimal periods, which all should line up and be in fixed positions. This checking is important because the new CIMIS format included some changes which caused the fixed format records to be altered in somewhat arbitrary fashion. A step in the processing of downloaded CIMIS data used to be manual editing of the data file to fill in incomplete records and force the commas and fields to line up properly. This step has been replaced with the FIXDASHV2 program which automates that editing. When PREFORMV2 is done, it issues a message "EOF on 8760 PROGRAM TERMINATED NORMALLY". The number 8760 indicates the line number last read in the input file. EOF stands for 'End of File'.

3.5. CIMPRO1V4.FOR

Command: CIMPRO1V4 CSS-YY.PRE CSS-YY.DET (User Manual pg20)

This is the first of two programs at the heart of the CIMIS data processing. CIMPRO1V4 primarily checks the basic data set for quality control flags and marks days that are unusable.

Subroutine LOOK4E checks for occurrence of "--" (two dashes) or "***" in a record. Their presence would indicate an error. If found, the program halts and writes out the record. Then ERRCHKA checks that the year and station ID remain constant, that days and hours progress correctly. After 24 hours of data are checked and read into arrays, a series of calls are made to CHECKQA for (standard deviation of wind direction). For each of these variables, the QC flag is checked. The actual checking occurs with logical function VALIDA. The solar radiation and precipitation QC fields are not checked.

The current QC flag checks for CIMPRO1V4 are as follows:

M,S,N – data value is rejected

I – data value rejected unless this is wind velocity, then accepted. In the case of wind velocity, sometimes the wind speed of 1 mile per hour is flagged with an I. This speed is the lowest possible reporting speed for CIMIS. Subsequent processing (in CIMPRO2V5) will make this a calm hour and assign it a velocity of zero.

R – this data is accepted. This program used to allow the user to set whether data with this flag is accepted or rejected. However, in order to simplify the procedures, we decided to hardwire in acceptance.

CIMIS (<http://www.cimis.water.ca.gov/cimis/dataQcCurrent.jsp>) lists the following flags for hourly data.

I	Data value has no meaning/ignore
M	Missing
S	Sensor is not in service or data is out of sensor threshold
A	Historical average
E, T	Historical average of one of the sensors used to calculate a parameter
N	Data value is not collected by this station
P	Quality test pending
Q	All quality control could not be performed because a comparison sensor is severe. This flag is not set when the comparison sensor is N
R	Data is far out of historical limits
Y	Data is moderately out of historical limits

Table 3.5.1. Cimic QC flag list.

Based on looking over data associated with R and Y flags, we determined that for the most part the values did not seem untoward. And to exclude R and Y values might result in throwing away usable data. For Q flags, there were some problems. Q seems to be primarily used for net radiation. The examples we found generally occurred when the solar radiation sensor was not working and there were numerous net radiation missing values or values flagged with an S. Consequently, the Q values were associated with days that would be rejected entirely. There were only a few P flags. Generally, it is difficult to write an algorithm that handles all of these situations correctly. Therefore, it is incumbent upon the user to determine what relevance or action to take should A,E,T,P,Q,R or Y flags are present in the data record. An easy way to locate these flags is to use Windows Explorer to search all *.txt files (the files that result after running UNIX2DOS) for the occurrence of, for example, ",Q,". Then opening any files listed in the search with, say, NOTEPAD, and searching for ",Q," to look at the data to decide what to do.

If more than 4 hours of any of the five variables (net radiation, air temperature, wind velocity, wind direction and sigma theta) are bad values within a day, then that day is rejected. CIMPRO1V4 places a "#" in column 105 indicating that the day is rejected. There is no further processing of that day in CIMPRO2V5.

CIMPRO1V4 produces two output files: the first is *filename*.DET, which contains the data that has been checked (along with any rejected days showing a "#" in column 105) and also a *filename*.SUM file, which provides a summary of bad values and rejected days. Any existing files with the same name are overwritten. The SUM file provides useful information about the data (Table 3.5.2). The input and output files are listed. Each record that had at least one instance of bad data (determined by checking the qc flags) is listed. The record number is provided along with the Julian Day and sensor type. A sequence of G and B are provided for each hour depending on whether that hour was good or bad. The number of bad hours is shown at the end of each day/sensor record. In the first line of Table 3.5.1, for example, the record number is 2256 which occurred on Julian day 94 and the sensor was air temperature. In this case, hours

```

CIMPRO1V4 ....
INPUT FILE: c86-05.pre
PROCESSED OUTPUT FILE c86-05.DET
SUMMARY FILE c86-05.SUM
ONLY DAYS WITH BAD OR MISSING VALUES ARE LISTED.
AN * ON LEFT INDICATES THAT ALL 24H ARE BAD FOR THAT SENSOR.
AN * ON RIGHT INDICATES THAT DAY WILL BE MARKED UNUSABLE.

RECORD DAY SENSOR      HOURLY FLAG : GOOD/BAD
-----
  2256  94 AIRTEMP      G G G G G G G G G G B B G G G G G G G G G G G G  2
  8640 360 AIRTEMP      G B B B B B B B B B B B B B B B B B B B B G G G G 19*
  8664 361 AIRTEMP      G G G G G G G G B G B G B G G G B B B G G G G G G  6*
  8688 362 AIRTEMP      G G G G G G G G G G G G G G G G G G G G G G G B B  2
*  8712 363 AIRTEMP      B B B B B B B B B B B B B B B B B B B B B B B B B 24*
  8736 364 AIRTEMP      B B B B B B B B B B B B B B B B B B B B B G G G B B 21*
*  8760 365 AIRTEMP      B B B B B B B B B B B B B B B B B B B B B B B B B 24*
CRITERIA FOR REJECTING DAYS: WHEN > 4 RECORDS ARE BAD FOR AT LEAST 1 VARIABLE
      8760 RECORDS EXAMINED WHICH IS          365 DAYS
GOOD DAYS=      360, BAD DAYS=          5, SUM OF BOTH=      365
R WAS NOT CONSIDERED AS BAD DATA IN THIS RUN

```

Table 3.5.2. SUM file for Tulare County, Lindcove Station #86 for 2005.

11 and 12 were flagged, giving a total of 2 hours. For the second line (8640 line number), there were 19 bad temperature records and since that exceeds 4 bad hours, the day will not be used in subsequent calculations. For the fifth record listed (8712 line number), all of the temperature measurements for that day were bad. Consequently, the asterisk to the left of this record indicates that all of the hours were bad for air temperature. The day will be rejected. At the bottom is a summary of good versus bad days.

The SUM file can also contain information about other errors which may arise in the processing of the PRE file; for example, if records are missing or out of sequence. In these cases, the record number is given and the original data file (TXT or PRE) file can be examined by going to that record number to determine what the problem is. CIMPRO1V4 does not fix all problems with the input data, but tries to bring attention to those problems in the SUM file. It is a good idea to always examine the SUM file to look for errors which may have occurred in the data set.

3.6. CIMPRO2V5.FOR

Command: CIMPRO2V5 CSS-YY.DET YYCIMSSS.2PC (User Manual pg21)

The major tasks for CIMPRO2V5 are to look for missing or bad hourly values in each usable day (days not flagged with '#' in column 105 by CIMPRO1V4) for net radiation, temperature, wind speed and direction, and wind direction standard deviation and, if necessary, to linearly interpolate for missing values. It sets wind speeds below 1.25 m/s

to zero to indicate calm conditions. It estimates stability class and corrects for negative net daytime radiation values. It reformats the meteorological data with the ISCST3 compatible portion on the right hand end of each record. The output file, which uses a 2PC extension, consists of the interpolated CIMIS data (if any) in English units, then appended on each hourly record, the ISCST3 reformatted record.

The first step for CIMPRO2V5 is to check column 105 for the '#' character, which connotes that the day is unusable (as determined in CIMPRO1V4) because 1 or more sensors had at least 5 or more missing values. If the day is unusable, CIMPRO2V5 writes out the 24 hourly records for that day and moves on without any calculations. Then for each sensor type, CIMPRO2V5 calls subroutine CHECKQ to determine which hourly values are valid and which are not. If there are invalid values, then it linearly interpolates between bracketing good values. For wind speed, if the speed is less than 1.25 m/s, it is set to zero in order to make that a calm hour (Johnson 2001, Appendix 2). Wind speeds changed to zero are flagged with a '@' in column 58, which is the QC flag for wind speed.

After filling in for the bad values, units are converted to metric, wind direction is reversed by 180 degrees, since ISCST3 requires the 'FROM' direction. Also, mixing height is set to 320m for both urban and rural components. The DOSTAB subroutine is called to calculate the stability class. Stability class is based on night/day (determined by net radiation, but see Section 3.6.4), standard deviation of wind direction and wind speed (Part 1, Vidrio and Johnson 2011, Johnson 2001).

3.6.1. Stability classification

Using guidelines set by USEPA (2000), an initial stability classification was determined by utilizing the standard deviation of wind direction. This initial stability classification was then modified depending on wind speed and night versus day. The final stability was determined by allowing no more than one stability class change per hour. For determining night and day, the net radiation was used. Negative net radiation was defined as night and positive is defined as day.

Established cutoff points for standard deviation of wind direction used to determine the initial stability classification do not entirely reflect the 2 meter height used at CIMIS stations, USEPA adjustment equations were used to modify cutoff points (Johnson, 2001). Adjusted cutoff points were determined to be 18.0, 16.2, 11.9, 7.9, 5.1 degrees (Table 3.6.1).

Table 3.6.1 Adjustment to wind direction standard deviation categories for roughness and 2 meter wind measurement height for CIMIS data (Johnson 2001, Table 4.4).

Stabi Clas	Lower Cutoff Bound (degree)	Roughness Factor	Wind Measurement Height	Adjusted Cutoff (degree)	
		$Z_0 = 3\text{cm}$	Factor $Z = 2\text{m}$		
		$\left(\frac{Z_0}{15}\right)^{0.2}$	P_θ	$\left(\frac{Z}{10}\right)^{P_\theta}$	
1	22.50	0.72	-0.06	1.10	18.0
2	17.50	0.72	-0.15	1.27	16.2
3	12.50	0.72	-0.17	1.31	11.9
4	7.50	0.72	-0.23	1.45	7.9
5	3.80	0.72	-0.38	1.84	5.1

3.6.2. Validation of CIMPRO2V3 determined stability

To confirm CIMPRO2V5 determined stability classifications, three random 24-hour periods were selected from the processed CIMIS data files and manual stability classifications were performed following the procedure listed in section 3.6.1. The results were compared to the CIMPRO2V5 determined stability classifications for the same 24-hour periods. Results are shown on Appendix 2. These results verified that CIMPRO2V3 correctly performed stability classifications because the randomly selected 24-hour files gave the identical stability classifications as those manually calculated.

3.6.3. Calms determination

The term “calm” is used to define the situation when the wind speed is so low that either speed or direction is indeterminate. For the CIMIS network, in which an hourly average is taken from 60 individual one minute measurements, the lowest possible hourly wind speed average is 0.447 m/s (Johnson, 2001). At hourly average speeds near 0.447 m/s, some fraction of the 60 measurements will be “calms”. Consistency with previous work, the CIMIS calms threshold is set to 1.25 m/s (Johnson, 2001). The program sets calm hour wind speeds to 0.0 m/s in the final record.

A small subset of the data was analyzed for percentage of calms. Specifically, one complete year of meteorological data was selected from every one of the 18 CIMIS weather stations in the newly created ISC database and analyzed for calm hour percentage using Microsoft® Excel’s PivotTables. It was found that in some of the year-long data sets, the percentage of calm hours exceeded 50% for the year. Although these percentages were sometimes far greater than the ARB recommended 10% limit, it does not preclude using portions of that data set for shorter simulations. Therefore, we opted to include all of the data sets that we processed. It is incumbent upon the user of this data to assess calms frequencies.

3.6.4. Negative net radiation

Stability classes EF are typical of nighttime conditions and classes ABC are typical of daytime conditions. In reviewing meteorological data for a modeling project, Barry

(personal communication) noticed nighttime stability classes occurring during the day in several instances. Further investigation of this showed that in some cases, negative net radiation values were found in the CIMIS data set during daytime hours. Since CIMPRO2V5 uses negative versus positive net radiation for determination of night/day, the occasional presence of negative net radiation during daytime hours led to night time stability assignments for daytime hours.

Table 3.6.4 provides an example of negative net radiation during the day. The Lindcove station is in Tulare County. Sunrise and sunset times for January 16, 2005 are 7:07 and 17:05 (U.S. Naval Observatory). Between those hours, one would not expect to see stability class 6 (=F). Solar radiation goes from 0 at 600, starts to increase, then goes to zero again at 1800. However, net radiation is negative except for 1400 and 1500. The quality control flags from CIMIS are asterisks, indicating normal values and no problems. CIMPRO2V4 (the version previous to CIMPRO2V5) has used nighttime conditions to evaluate the stabilities for these daytime hours with the result that hours 800-1300 receive a stability class assignment of 6, which is nighttime stable conditions and would not apply during daylight hours. The algorithm for determining stability is required to change the hourly stability by no more than 1 class every hour. With positive net radiation in hours 1400 and 1500, the algorithm begins to move the stability towards daylight conditions, but the onset of negative net radiation at hour 1600 causes the algorithm to reverse and head back towards nighttime stability.

Table 3.6.4. Example of negative net radiation from CIMIS data set. Station 86, Lindcove (Tulare County) January 16, 2005. Jul is Julian Day, Sol Rad is solar radiation, qc is the quality control code for the following field, Net Rad is net radiation, Stability Class is the resulting stability class as evaluated by CIMPRO2V4 (A=1, B=2....F=6). Shaded cells are suspect due to negative net radiation during the daytime.

Stn Id	Station	Date	Hour	Jul	qc	Sol Rad (Ly/day)	qc	Net Rad (Ly/day)	Stability Class, CIMPRO2V4
86	Lindcove	1/16/2005	100	16	*	0	*	-35	6
86	Lindcove	1/16/2005	200	16	*	0	*	-35	6
86	Lindcove	1/16/2005	300	16	*	0	*	-35	6
86	Lindcove	1/16/2005	400	16	*	0	*	-35	6
86	Lindcove	1/16/2005	500	16	*	0	*	-35	6
86	Lindcove	1/16/2005	600	16	*	0	*	-35	6
86	Lindcove	1/16/2005	700	16	*	0	*	-35	6
86	Lindcove	1/16/2005	800	16	*	8	*	-29	6
86	Lindcove	1/16/2005	900	16	*	73	*	-23	6
86	Lindcove	1/16/2005	1000	16	*	127	*	-7	6
86	Lindcove	1/16/2005	1100	16	*	108	*	-39	6
86	Lindcove	1/16/2005	1200	16	*	130	*	-32	6
86	Lindcove	1/16/2005	1300	16	*	159	*	-11	6
86	Lindcove	1/16/2005	1400	16	*	381	*	155	5
86	Lindcove	1/16/2005	1500	16	*	252	*	80	4
86	Lindcove	1/16/2005	1600	16	*	71	*	-29	5
86	Lindcove	1/16/2005	1700	16	*	20	*	-20	6
86	Lindcove	1/16/2005	1800	16	*	0	*	-35	6
86	Lindcove	1/16/2005	1900	16	*	0	*	-35	6
86	Lindcove	1/16/2005	2000	16	*	0	*	-35	6
86	Lindcove	1/16/2005	2100	16	*	0	*	-35	6
86	Lindcove	1/16/2005	2200	16	*	0	*	-35	6
86	Lindcove	1/16/2005	2300	16	*	0	*	-35	6
86	Lindcove	1/16/2005	2400	16	*	0	*	-35	6

As far as we can determine, there does not appear to be good physical reason for daytime negative net radiation values. CIMIS personnel were unsure as to why the net radiation was negative during the day (Temesgen 2013). The first step was to develop criteria for determining day and night. The U.S. Naval Observatory web site was queried to produce sunrise and sunset times for Redding and San Diego, two cities that span most of California north to south. I used these two sets to produce a combined sunrise/sunset data table (Table 3.6.5). Where hours differed between the cities, I chose the station which gave the longest night time period, which was Redding for sunrise and San Diego for sunset. This provided a slight conservatism because night time stability classes produce higher concentrations for a given flux.

Table 3.6.5 was encoded into a function (FIXNETRAD) which is included in the CIMPRO2V5 listing. Nighttime hours for January, for example, are defined as hours from midnight to 9AM and from 1600 (4PM) to midnight. Otherwise hours are daytime and net radiation is assumed to be positive. Real Function FIXNETRAD was tested on isolated cases and on whole data sets to verify correct operation.

Table 3.6.5. Monthly median sunrise/sunset times for San Diego and Redding, California for 2013 (U.S. Naval Observatory) and assignment of daylight hour screen for function. Table shows calculation of 1 hour after sunrise and 1 hour before sunset. The USEPA defines night time as including 1 hour after sunrise and 1 hour before sunset. The function FIXNETRAD overrides negative net radiation values during daytime hours in order to calculate daytime stabilities.

Month	Medians from each month				Calculate Last Hour of Night in Morning and First Hour of Night in Evening			
	Sunrise	Sunrise	Sunset	Sunset	Redding Sunrise		San Diego Sunset	
	San Diego	Redding	San Diego	Redding	Redding, 1 hour after sunrise	Last hour of night	San Diego, 1 hour before sunset	First hour of night
1	651	731	1707	1709	831	9	1607	16
2	632	704	1734	1745	804	8	1634	17
3	557	619	1758	1818	719	7	1658	17
4	518	530	1820	1850	630	6	1720	17
5	449	452	1841	1921	552	6	1741	18
6	441	439	1858	1943	539	6	1758	18
7	452	453	1857	1938	553	6	1757	18
8	513	521	1832	1905	621	6	1732	18
9	533	552	1754	1817	652	7	1654	17
10	554	622	1714	1727	722	7	1614	16
11	620	657	1647	1652	757	8	1547	16
12	644	727	1645	1644	827	8	1545	16

Columns 001:0056														
1			2			3			4			5		
123456789012345678901234567890123456789012345678901234567890123456														
0080,09-01-01,01,001,*,						0.000,*,								
0080,09-01-01,02,001,*,						0.000,*,								
0080,09-01-01,03,001,*,						0.000,*,								
Columns 0057:118														
6			7			8			9			10		
78901234567890123456789012345678901234567890123456789012345678														
,*,						3.200,*,								
,*,						2.900,*,								
,@,						2.300,*,								
Columns 119:158														
11			12			13			14			15		
901234567890123456789012345678901234567890123456789012345678														
09010101						279.2000								
09010102						282.5000								
09010103						255.4000								

Table 3.6.6 Three lines of processed CIMIS data from Station 80 for 2009. The first two lines are column counters and not part of the 2PC file. The information is wrapped in order to fit it into the text box. In this example, column 105 does not show a “#”. A “#” in column 105 connotes an unusable day. The wind speed for hour 3 was less than 1.25 m/s, hence there is a ‘@’ marker in column 58.

The final result is output into the *.2PC file and consists of original CIMIS record, with potential modifications for interpolation with the ISCST3 compatible met data appended to it. The convention utilized in Part 1 of this document (Vidrio and Johnson 2011) is to use a 2PC extension for this data file which contains both the processed raw CIMIS data and appended ISCST3 formatted/calculated data. An extract from a 2PC file contains three records each with 158 columns in a fixed-field format (Table 3.6.6). Columns 1-118 contained the processed original CIMIS data and any interpolated values calculated by CIMPRO2V5. Columns 119-to 158 contain the ISCST3 ready meteorological data which includes the estimated stability class and fixed mixing height values.

The explicit field and column definitions for Table 3.6.6 and for the 2PC (the output from CIMPRO2V5) file are provided in Table 3.6.7.

CIMIS portion after processing with CIMPRO2V5	
Station Number	001:004
Year	006:007
Month	009:010
Day	012:013
Hour	015:016
Julian Day	018:020
Solar Radiation(Ly)	024:032
Net Radiation (Ly)	036:044
Temperature (F)	048:056
Wind Speed (mph)	060:068
Direction (From, deg)	072:080
Std Dev Direction	084:092
Precipitation (in)	096:104
Day usability flag	105:105
ISCST3 portion	
Year	111:112
Month	113:114
Day	115:116
Hour	117:118
Wind Dir (To)	119:127
Wind Speed (m/s)	128:136
Temperature (K)	137:142
Stability	144:144
Rural Mix Ht (m)	145:151
Urban Mix Ht (m)	152:158

Table 3.6.7 Field and column definitions for *.2PC file produced by CIMPRO2V5.

3.7. MAKEISCV2.FOR

Command: MAKISCV2 YYCIMSSS.2PC YYCIMSSS.ISC (User Manual pg22)

The principal task for MAKISCV2.FOR is to extract the ISC ready meteorological data from the 2PC file and put it into a separate file, ready for use with ISC. In addition it creates a first line consisting of 4 numerical values which represent, respectively, the surface meteorological station, year, upper air meteorological station and year (Table 3.7). The surface meteorological station number is constructed by appending the 3 digit CIMIS station number to the digits, '11'. The upper air station number is meaningless. However, in the ISCST3 control file, these two station numbers and the year must be correctly referenced.

11080	09	11111	09				
09010101	279.2000	1.4305	278.5	4	320.0	320.0	
09010102	282.5000	1.2964	278.5	5	320.0	320.0	
09010103	255.4000	0.0000	278.6	6	320.0	320.0	

Table 3.7. Extract from C80-09.ISC file.

3.8. Program Listings

Appendix 1 contains complete listings for the six FORTRAN programs described. The older programs represent accumulated modifications over almost a 20 year period. As such, they include unused code and perhaps less readable coding, which in theory, should be deleted or reformatted for readability. Cleaning up the code is itself a separate task for another time. The programs have been tested throughout the years by comparing manual calculations to computed results. In some places the program listing may wrap the text line, which does not actually occur in the code.

4. Summary

This document describes a series of seven programs consisting of one Turbo C program and six FORTRAN programs used to convert downloaded CIMIS meteorological data into ISCST3 ready meteorological data sets. This document is intended as a companion document to Part 1 (Vidrio and Johnson 2011), which details how to use these programs. These programs reformat the raw CIMIS data, check for missing values and interpolate where possible, then calculate stability and reformat the key values for use with ISCST3.

5. References

Barry, Terrell. 2013. Personal communication.

California Environmental Protection Agency. 2010. Kettleman City Community Exposure Assessment Work Plan. California Environmental Protection Agency: Air Resources Board, Department of Pesticide Regulation, Department of Toxic Substances Control, Office of Environmental Health Hazard Assessment, State Water Resources Control Board June 17, 2010.

<http://www.cdpr.ca.gov/docs/emon/pubs/protocol/study267protocol.pdf>

Johnson, Bruce. 2001. Evaluating the Effectiveness of Methyl Bromide Soil Buffer Zones in Maintaining Acute Exposures below a Reference Air Concentration. California Environmental Protection Agency, Department of Pesticide Regulation, EH00-10 April 2001. Available at: <http://www.cdpr.ca.gov/docs/emon/pubs/ehapreps/eh0010.pdf>

USEPA . 2000. Meteorological Monitoring Guidance for Regulatory Modeling Applications. United States Environmental Protection Agency, Office of Air Quality Planning and Standards, Research Triangle Park, NC 27711. EPA-454/R-99-005 February 2000.

Temesgen, Bekele. 2013. Email to Bruce Johnson on “RE: Negative daytime net radiation” dated October 1, 2013.

U.S. Naval Observatory. Data Services. aa.usno.navy.mil/data/ (last accessed Jan 3, 2014)

Vidrio, Edgar and Bruce Johnson. 2011. Part 1: DATABASE CREATION OF ISCST3-READY FILES PREPARED FROM CIMIS METEOROLOGICAL NETWORK DATA. July, 2011. (rev. Jan 2014) California Environmental Protection Agency. Department of Pesticide Regulation. Environmental Monitoring Branch . Air Program. P.O. Box 4015 . Sacramento, California 95812-4015

APPENDIX 1: PROGRAM LISTINGS

UNIXORDOSV2.FOR

```
C Last change: BJ 21 Jun 2011 3:39 pm
PROGRAM UNIXORDOSV2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C UNIX STYLE LINE TERMINATION IS LF (LINE FEED, ASCII 10, OR HEX 0A) ONLY
C DOS STYLE LINE TERMINATION IS CR (CARRIAGE RETURN, ASCII 13, OR HEX 0D), AND LF (CR LF)
C
C LF IS ALSO SHOWN AS ^J (CNTRL J)
C CR IS ALSO SHOWN AS ^M (CNTRL M)
C
C THIS IS SECOND VERSION, PUT IN DETERMINATION OF DOS OR UNIX STYLE
C
C NOTE: SPACE = DECIMAL 32
C PRINTABLE CHARACTERS DECIMAL 33 TO DECIMAL 126
C CR IS DECIMAL 13
C LF IS DECIMAL 10
C
C THIS PROGRAM DETERMINES IF FILE USES 0D0A LINE TERMINATORS (DOS STYLE)
C OR 0A LINE TERMINATORS (UNIX STYLE). IT GIVES YES OR NO ANSWER
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
IMPLICIT NONE
CHARACTER*50 FNAME
CHARACTER*10000 CHUNK !CHUNK OF FILE TO LOOK FOR SPEC CHARS
CHARACTER*1 CHUNK2(10000),CHUNK3(10000)
CHARACTER*1 LF,CR
INTEGER I ,J,K,M ,L
INTEGER FILEBYTES
LOGICAL FEXIST
INTEGER COUNTLF,COUNTCR,COUNTCRLF,COUNTCRCR
INTEGER COUNTSPACES,COUNTPRINTABLE
INTEGER BOUND
LF=CHAR(10) !LINEFEED
CR=CHAR(13) !CARRIAGE RETURN
CALL GETCL(FNAME)
IF(LEN_TRIM(FNAME).EQ.0)THEN
WRITE(6,100)
100 FORMAT(1X,'UNIXORDOSV2: DETERMINE IF FILE HAS UNIX OR DOS',
1 /13X,' STYLE LINE TERMINATORS',
1 /1X,'ENTER NAME OF FILE TO CHECK: ')
READ(5,150)FNAME
150 FORMAT(A50)
ENDIF

INQUIRE(FILE=FNAME,FLEN=FILEBYTES,EXIST=FEXIST)
IF(.NOT.FEXIST)THEN
WRITE(6,43)FNAME
43 FORMAT(1X,'CANNOT FIND ',A50,/1X,'CHECK THE SPELLING ')
WRITE(6,44)
44 FORMAT(1X,'UNIXORDOSV2 IS STOPPING ')
STOP
ENDIF

OPEN(UNIT=1,STATUS='OLD',ACCESS='TRANSPARENT',FILE=FNAME)
!INITIALIZE STRING
DO I=1,10000
CHUNK(I:I)=CHAR(0)
END DO
!GET THE FIRST 10000 BYTES
READ(1,200)CHUNK
200 FORMAT(A10000)
DO I=1,10000
CHUNK2(I)=CHUNK(I:I)
CHUNK3(I)=CHUNK2(I)
IF(ICHAR(CHUNK3(I)).EQ.10)CHUNK3(I)='@'
IF(ICHAR(CHUNK3(I)).EQ.13)CHUNK3(I)='@'
END DO
```

```

!PROCESS THE ARRAY HERE
BOUND=MIN(FILEBYTES,10000)
!LOOK FOR INSTANCES OF 0A0D, AND 0A AND 0D SEPARATELY AND 0D0D
COUNTLF=0
COUNTCR=0
COUNTCRLF=0
COUNTCRCR=0
COUNTSPACES=0
COUNTPRINTABLE=0
DO I=1,BOUND

K=ICHAR(CHUNK2(I))
IF(K.EQ.32)COUNTSPACES=COUNTSPACES+1
IF(K.GE.33.AND.K.LE.126)COUNTPRINTABLE=COUNTPRINTABLE+1
IF(K.EQ.13)COUNTCR=COUNTCR+1
IF(K.EQ.10)COUNTLF=COUNTLF+1
IF(L.LT.BOUND)THEN
  L=ICHAR(CHUNK2(I+1))
  IF(K.EQ.13.AND.L.EQ.10)COUNTCRLF=COUNTCRLF+1
  IF(K.EQ.13.AND.L.EQ.13)COUNTCRCR=COUNTCRCR+1
ENDIF

END DO

OPEN(UNIT=2,STATUS='UNKNOWN',FILE='UORD.TMP') !JUNK FILE JUST IN CASE
DO I=1,500

  K=20*(I-1)
  ! WRITE(6,300)(CHUNK3(M),M=K+1,K+20)
  WRITE(2,300)(CHUNK3(M),M=K+1,K+20)
300   FORMAT(1X,20(2X,A1))

  ! WRITE(6,305)(ICHAR(CHUNK2(M)),M=K+1,K+20)
  WRITE(2,305)(ICHAR(CHUNK2(M)),M=K+1,K+20)
305   FORMAT(1X,20(I3))

  ! WRITE(6,310)(ICHAR(CHUNK2(M)),M=K+1,K+20)
  WRITE(2,310)(ICHAR(CHUNK2(M)),M=K+1,K+20)
310   FORMAT(1X,20(1X,Z2))

END DO
WRITE(2,400)FILEBYTES
400  FORMAT(1X,I10,' BYTES IN FILE ')
WRITE(2,405)BOUND
405  FORMAT(1X,'THE TOTAL CHARACTERS EXAMINED WAS ',I6)
WRITE(2,500)COUNTSPACES,COUNTPRINTABLE
500  FORMAT(1X,'THERE WERE ',I6,' SPACES'/1X,'AND',8X,I6,
1     ' PRINTABLE CHARACTERS')
WRITE(2,505)COUNTCR,COUNTLF
505  FORMAT(1X,'THERE WERE ',I6,' CARRIAGE RETURNS AND ',
1     I6,' LINE FEEDS')
WRITE(2,510)COUNTCRLF,COUNTCRCR
510  FORMAT(1X,'THERE WERE ',I6,' CRLF AND ',I6,' CRCR ')
IF(COUNTCRCR.EQ.0)THEN
  IF(ABS(COUNTCR-COUNTLF).LE.1)THEN
    WRITE(6,520)FNAME
    WRITE(2,520)FNAME
520  FORMAT(1X,'THE FILE: ',A30,
1     /1X,' HAS NORMAL DOS FORMATTING',
2     /1X,' THERE IS NO NEED TO RUN UNIX2DOS')
  ELSEIF(COUNTCR.EQ.0.AND.COUNTLF.GT.0)THEN
    WRITE(6,525)FNAME
    WRITE(2,525)FNAME
525  FORMAT(1X,'THE FILE: ',A30,
1     /1X,' APPEARS TO BE UNIX FORMATTED ',
2     /1X,' YOU NEED TO RUN THE UNIX2DOS PROGRAM ON IT ')
  ENDIF
ELSEIF(COUNTCRCR.GT.0)THEN

```

```
WRITE(6,530)FNAME
WRITE(2,530)FNAME
530  FORMAT(/1X,'THE FILE: ',A30,
1     /1X,'APPEARS TO BE CORRUPTED POSSIBLY '
1     /1X,'BECAUSE YOU RAN UNIX2DOS ON A DOS FILE... ',
1     //1X,'DOWNLOAD AGAIN FROM CIMIS AND RUN UNIXORDOSV2 ',
1     /1X,'ON THE DOWNLOAD BEFORE DOING ANYTHING ELSE. ')
ENDIF
END PROGRAM
```

FIXDASHV2.FOR

```
C      Last change:  BJ      7 Jul 2011      1:31 pm
      PROGRAM FIXDASHV2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C THIS PROGRAM WRITTEN AT REQUEST OF TB TO
C AUTOMATE FIXING THE MISSING VALUES IN CIMIS DATA
C FILES WHICH ARE "--" AND THE FIELD IS PACKED, SO
C THAT THE FIXED FORMATTING IS MESSED UP
C
C [110706 PARSEIT SEEMS TO BE WORKING CORRECTLY, NEXT
C PROCESS TO INSERT AND PAD OUT THE FIELDS TO CONSTRUCT
C A CORRECTED LINE WITH MISSING VALUE CODES]
C
C [110707 APPEARS TO BE WORKING, VERSION 2 ADDS ON
C INPUT/OUTPUT STYLE OF RUNNING THE PROGRAM TO BE
C CONSISTENT WITH THE OTHER PROGRAMS USED IN CIMIS MET PROCESSING
C
C I RAN 84 FILES THAT CONTAINED MISSING VALUES THROUGH FIXDASHV2
C AND THEN PREFORMV2. TWO FILES BOMBED BECAUSE INSTEAD OF '--'
C IN THE FIELD WHICH WAS MARKED MISSING, THERE WAS ##### AND PREFORMV2
C BOMBED WHEN IT READ THE VALUE (C125-05 AND C161-06)
C THAT ONLY HAPPENED FOR TWO FILES, AND ONLY IN THE SOLAR RADIATION VARIABLE,
C SO I ADDED A CHECK FOR JUST THAT SITUATION, THE QC GETS CHANGED FROM P TO M
C AND THE VALUE IS CHANGED TO -99 AS WITH THE DASHES CASE.
C ALSO, THERE WAS ON FILE WHICH EVIDENTLY CONTAINED TEXT FOR THE HEADERS IN THE FIRST
C LINE, WHICH WONT WORK AND ALSO ANOTHER FILE THAT CONTAINED TEXT FOR THE
C STATION NAME, WHICH ALSO WONT WORK, THIS STUFF CAN BE FOUND IN
C PARLIER/CIMIS/FIX-MISSING/DASHFILES/REPORT.OUT, REPORTPRE.OUT, THE 84 FILES WERE
C OBTAINED FROM EDGAR'S WORK AND WERE SELECTED BY SEARCHING FOR '--'
C SINCE PREFORM RAN CORRECTLY ON MOST OF THEM, AND BASED ON MY DEVELOPMENT TESTING,
C I CONCLUDE THAT FIXDASHV2 IS WORKING CORRECTLY
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT NONE
      CHARACTER*91 LINE,NEWLIN
      CHARACTER*2 DD
      CHARACTER*5 SOLRAD,NETRAD
      CHARACTER*7 AIRTEMP,WINDSP,WINDDIR,STDEV
      CHARACTER*8 PRECIP
      CHARACTER*1 QSOLRAD,QNETRAD,QAIRTEMP,QWINDSP
      CHARACTER*1 QWINDDIR,QSTDEV,QPRECIP
      INTEGER I,J,K,L,M
      INTEGER LCOUNT,TOTDASHCOUNT
      !INTEGER CURFIELD,LENCURF(7) !KEEPS TRACK OF WHICH FIELD WORKING ON, AND LENGTH
OF THAT FIELD
      !DATA LENCURF/5,5,7,7,7,7,8/
      CHARACTER*8 MISVAL(3)
      DATA MISVAL/' -99',' -99.0',' -99.00' / !THREE FORMS OF MISSING VALUE

      DATA DD /'--'/

CCCCCCCCCCCC INPUT/OUTPUT MODULES USED IN OTHER PROGRAMS IN THIS SUITE CCCCCCCCCCCCCCCCCC
      CHARACTER*35 FIN,FOUT
      LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
      INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C      OPEN(UNIT=1,STATUS='OLD',FILE='TEST.TXT')
C      OPEN(UNIT=2,STATUS='UNKNOWN',FILE='TEST.OUT')
C      OPEN(UNIT=3,STATUS='UNKNOWN',FILE='FINPROD.OUT')

      UIN=1
      UOUT=2
      CALL IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)

      !READ FIRST LINE AND CHECK FOR '--', IF NONE, THEN
```



```

!GO TO NEXT LINE, IF AT LEAST ONE, THEN PROCESS
LCOUNT=0
TODDASHCOUNT=0
1 CONTINUE
  READ(1,50,END=10000)LINE
50  FORMAT(A91)
  LCOUNT=LCOUNT+1
C   WRITE(2,51)LINE,LCOUNT
C   WRITE(6,51)LINE,LCOUNT
C1  FORMAT(1X,A91,/1X,'LCOUNT= ',I4)
  J=INDEX(LINE,DD)
  IF(J.EQ.0)THEN
    WRITE(2,70)LINE
    !WRITE(3,70)LINE
70  FORMAT(A91)
  ELSE ! OK FOUND MISSING VALUE, PARSE LINE, AND REBUILD NEW LINE

  CALL PARSEIT(LINE,SOLRAD,NETRAD,AIRTEMP,WINDSP,WINDDIR,
*          STDEV,PRECIP,QSOLRAD,QNETRAD,QAIRTEMP,QWINDSP,
*          QWINDDIR,QSTDEV,QPRECIP)

C   WRITE(2,10010)LINE,QSOLRAD,SOLRAD,QNETRAD,NETRAD,
C   *   QAIRTEMP,AIRTEMP,QWINDSP,WINDSP,
C   *   QWINDDIR,WINDDIR,QSTDEV,STDEV,
C   *   QPRECIP,PRECIP
C10010  FORMAT(1X,A91/1X,2(A1,1X,A5),2(a1,1X,A7)/1X,2(A1,1X,A7),
C   *   /1X,A1,1X,A8)
  !NOW GO THROUGH LINE, FIELD BY FIELD, DETERMINE WHERE MISSING VALUES AND SET UP
  !FIELD FOR REBUILDING, ALSO FORCE QC CODE TO M IF MISSING

  DO I=1,91      !BLANK THE NEW LINE
    NEWLIN(I:I)=' '
  END DO
  !I'M ADOPTING THE POLICY THAT IF THE FIELD CONSISTS OF --, THEN THE QC FLAG
SHOULD BE M
  !I'VE SEEN CASES WHERE IT'S E AND NOT M, SO WILL CHANGE TO M FOR MISSING
  !FIRST 25 CHARACTERS ARE HOPFULLY ALWAYS GOOD
  NEWLIN(1:25)=LINE(1:25)

  !CURFIELD=1 !SOLRAD
  IF(INDEX(SOLRAD,DD).GT.0)THEN !IF POSITIVE THEN THIS IS MISSING
    QSOLRAD='M' !SET QC TO M
    SOLRAD(1:5)=MISVAL(1)(1:5)
    TODDASHCOUNT=TODDASHCOUNT+1
  ENDIF
C125-05 EG IF(SOLRAD(1:5).EQ.'#####')THEN !THIS HANDLES RARE INSTANCE OF THIS ANOMOLY
    QSOLRAD='M'
    SOLRAD(1:5)=MISVAL(1)(1:5)
  ENDIF
  IF(INDEX(NETRAD,DD).GT.0)THEN
    QNETRAD='M'
    NETRAD(1:5)=MISVAL(1)(1:5)
    TODDASHCOUNT=TODDASHCOUNT+1
  ENDIF
  IF(INDEX(AIRTEMP,DD).GT.0)THEN
    QAIRTEMP='M'
    AIRTEMP(1:7)=MISVAL(2)(1:7)
    TODDASHCOUNT=TODDASHCOUNT+1
  ENDIF
  IF(INDEX(WINDSP,DD).GT.0)THEN
    QWINDSP='M'
    WINDSP(1:7)=MISVAL(2)(1:7)
    TODDASHCOUNT=TODDASHCOUNT+1
  ENDIF
  IF(INDEX(WINDDIR,DD).GT.0)THEN
    QWINDDIR='M'
    WINDDIR(1:7)=MISVAL(2)(1:7)
    TODDASHCOUNT=TODDASHCOUNT+1
  ENDIF

```

```

IF (INDEX (STDEV,DD) .GT.0) THEN
  QSTDEV= 'M'
  STDEV(1:7)=MISVAL(2)(1:7)
  TOTDASHCOUNT=TOTDASHCOUNT+1
ENDIF
IF (INDEX(PRECIP,DD) .GT.0) THEN
  QPRECIP='M'
  PRECIP(1:8)=MISVAL(3)(1:8)
  TOTDASHCOUNT=TOTDASHCOUNT+1
ENDIF
!NOW FINISH CONSTRUCTING NEW LINE
NEWLIN(26:27)=QSOLRAD//','
NEWLIN(28:35)=SOLRAD(1:5)//','//QNETRAD//','
NEWLIN(36:43)=NETRAD(1:5)//','//QAIRTEMP//','
NEWLIN(44:53)=AIRTEMP(1:7)//','//QWINDSP//','
NEWLIN(54:63)=WINDSP(1:7)//','//QWINDDIR//','
NEWLIN(64:73)=WINDDIR(1:7)//','//QSTDEV//','
NEWLIN(74:83)=STDEV(1:7)//','//QPRECIP//','
NEWLIN(84:91)=PRECIP(1:8)
!WRITE OUT SPANKING NEW LINE
C      WRITE(2,300)LINE,NEWLIN
C300   FORMAT(1X,'COMPARE OLD AND NEW LINES',/1X,A91/1X,A91)
      WRITE(2,70)NEWLIN
      ENDIF
      GOTO1
10000  CONTINUE
      WRITE(6,12000)FIN(1:15),FOUT(1:15),LCOUNT,TOTDASHCOUNT
12000  FORMAT(1X,A15,1X,A15,'LINES= ',I5,' MISSING= ',I5)
      STOP
      END PROGRAM
      SUBROUTINE PARSEIT(LINE,SOLRAD,NETRAD,AIRTEMP,WINDSP,WINDDIR,
*          STDEV,PRECIP,QSOLRAD,QNETRAD,QAIRTEMP,QWINDSP,
*          QWINDDIR,QSTDEV,QPRECIP)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE PARSEIT CUTS UP LINE BASED ON COMMAS AND PLACES FIELDS INTO
C VARIABLES AS NAMED
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CHARACTER*91 LINE
      CHARACTER*2 DD
      CHARACTER*5 SOLRAD,NETRAD
      CHARACTER*7 AIRTEMP,WINDSP,WINDDIR,STDEV
      CHARACTER*8 PRECIP
      CHARACTER*1 QSOLRAD,QNETRAD,QAIRTEMP,QWINDSP
      CHARACTER*1 QWINDDIR,QSTDEV,QPRECIP
      INTEGER I,J,K,L,M
      INTEGER COMMPOS(17),CP(17) !SHOUL BE 17 COMMAS
      INTEGER COUNTCOMMA
      EQUIVALENCE (COMMPOS,CP)
      COUNTCOMMA=0
      DO I=1,91
        IF(LINE(I:I).EQ.' ') THEN
          COUNTCOMMA=COUNTCOMMA+1
          COMMPOS(COUNTCOMMA)=I
        ENDIF
      END DO
      IF(COUNTCOMMA.NE.17) THEN
        WRITE(6,100)COUNTCOMMA,LINE
100    *   FORMAT(1X,'ERROR FROM SUBROUTINE PARSE: WRONG # COMMAS ',I7,
*          /1X,'LINE IS ',/1X,A91)
        STOP
      ENDIF
      !BLANK THE FIELDS
      DO I=1,5
        SOLRAD(I:I)= ' '
        NETRAD(I:I)= ' '
      END DO
      DO I=1,7
        AIRTEMP(I:I)= ' '
        WINDSP(I:I)= ' '

```

```

        WINDDIR(I:I)=' '
        STDEV(I:I)=' '
    END DO
    DO I=1,8
        PRECIP(I:I)=' '
    END DO
    QSOLRAD=' '
    QNETRAD=' '
    QAIRTEMP=' '
    QWINDSP=' '
    QWINDDIR=' '
    QSTDEV=' '
    QPRECIP=' '
    !BRUTE FORCE PARSE RIGHT HERE
    QSOLRAD=LINE(CP(4)+1:CP(4)+1)
    K=CP(6)-CP(5)-1
    SOLRAD(1:K)=LINE(CP(5)+1:CP(6)-1)
    QNETRAD=LINE(CP(6)+1:CP(6)+1)
    K=CP(8)-CP(7)-1
    NETRAD(1:K)=LINE(CP(7)+1:CP(8)-1)
    QAIRTEMP=LINE(CP(8)+1:CP(8)+1)
    K=CP(10)-CP(9)-1
    AIRTEMP(1:K)=LINE(CP(9)+1:CP(10)-1)
    QWINDSP=LINE(CP(10)+1:CP(10)+1)
    K=CP(12)-CP(11)-1
    WINDSP(1:K)=LINE(CP(11)+1:CP(12)-1)
    QWINDDIR=LINE(CP(12)+1:CP(12)+1)
    K=CP(14)-CP(13)-1
    WINDDIR(1:K)=LINE(CP(13)+1:CP(14)-1)
    QSTDEV=LINE(CP(14)+1:CP(14)+1)
    K=CP(16)-CP(15)-1
    STDEV(1:K)=LINE(CP(15)+1:CP(16)-1)
    QPRECIP=LINE(CP(16)+1:CP(16)+1)
    M=LEN_TRIM(LINE)
    K=M-CP(17)
    PRECIP(1:K)=LINE(CP(17)+1:M)
    RETURN
    END SUBROUTINE
    SUBROUTINE IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CALLING PROGRAM NEEDS TO DECLARE THE FOLLOWING
C-----
C CHARACTER*35 FIN,FOUT
C LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
C INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
C UIN=1
C UOUT=2
C-----
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
    INTEGER UIN,UOUT
    CHARACTER*35 FIN,FOUT
    LOGICAL INSTATE,OUTSTATE,INSTATUS
    CHARACTER*70 CLINE

    CALL GETCL(CLINE)
    CALL IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)

    IF(INSTATE.AND.OUTSTATE)THEN
C        WRITE(6,500)FIN,FOUT,K1,K2
C500    FORMAT(1X,'IOREADY: INPUT,OUTPUT',/1X,A35,/1X,A35,
C        * /1X,'K1,K2 ',2I3)
        CONTINUE !STUFF ABOVE FOR DEBUGGING
    ELSEIF (.NOT.INSTATE)THEN
        WRITE(6,505)FIN,K1
    505    FORMAT(1X,'IOREADY: INPUT FILENAME ERROR ',A35,1X,I3)
        STOP
    ELSEIF (.NOT.OUTSTATE)THEN
        WRITE(6,510)FOUT,K2
    510    FORMAT(1X,'IOREADY: OUTPUT FILENAME ERROR ',A35,1X,I3)

```

```

        STOP
    ENDIF
    !CHECK TO MAKE SURE THE INPUT FILE EXISTS
    INQUIRE(FILE=FIN,EXIST=INSTATUS)
    IF(.NOT.INSTATUS)THEN
255      WRITE(6,255)FIN
        *      FORMAT(1X,'IOREADY: INPUT FILE NOT EXIST',
                /1X,A35, /1X,'PROGRAM STOPPING ')
        STOP
    ENDIF
    OPEN(UNIT=UIN,STATUS='OLD',FILE=FIN,ERR=1000)
    OPEN(UNIT=UOUT,STATUS='UNKNOWN',FILE=FOUT,ERR=2000)
    RETURN

    !ERRORS ON OPENING GO BELOW
1000   CONTINUE
        WRITE(6,1100)FIN
1100   FORMAT(1X,'ERROR TRYING TO OPEN IN FILE ',/1X,A35)
        STOP
2000   CONTINUE
        WRITE(6,2100)FOUT
2100   FORMAT(1X,'ERROR TRYING TO OPEN OUT FILE ',/1X,A35)
        STOP
    END SUBROUTINE

    SUBROUTINE IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)
    IMPLICIT NONE
    CHARACTER*35 FIN,FOUT
    CHARACTER*70 CLINE
    INTEGER K1,K2,IBLANK,K,I,L
    LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
    !K1, K2 ARE LENGTHS OF THE FILENAMES

    DO I=1,35
        FIN(I:I)=' '
        FOUT(I:I)=' '
    END DO
    !ASSUME CLINE HAS 1 OR MORE BLANK SPACES WHICH DIVIDE THE TWO FILENAMES
    !ASSUME THAT INPUT FILE IS THE FIRST FILE, AND OUTPUTFILE IS THE SECOND ONE
    K=LEN_TRIM(CLINE) !FIND OUT TOTAL STRING LENGTH
    !LOOK FOR FIRST BLANK
    IBLANK=0
    DO I=1,K
        IF(CLINE(I:I).EQ.' ')THEN
            IBLANK=I
            GOTO100
        ELSE
            CONTINUE
        ENDIF
    END DO
100    IF (IBLANK.EQ.0) THEN
        INSTATE=.FALSE.
        OUTSTATE=.FALSE.
        K1=0
        K2=0
        WRITE(6,200)CLINE
200    FORMAT(1X,'COMMAND LINE LACKING 2 FILENAMES ',/1X,A70)
        RETURN
    ELSEIF (IBLANK.GE.70)THEN
        WRITE(6,250)CLINE
250    FORMAT(1X,'FILENAME TOO LONG ',/1X,A70)
        K1=IBLANK
        K2=0
        INSTATE=.TRUE.
        OUTSTATE=.FALSE.
    ELSE
        FIN(1:IBLANK-1)=CLINE(1:IBLANK-1)
        L=K-(IBLANK-1)
        FOUT(1:L)=CLINE(IBLANK:K)
        FOUT(1:L)=ADJUSTL(FOUT(1:L))
        K2=LEN_TRIM(FOUT(1:L))
    
```

```

IF(K2.EQ.0)THEN
  INSTATE=.TRUE.
  OUTSTATE=.FALSE.
  RETURN
ENDIF
K1=IBLANK-1
INSTATE=.TRUE.
OUTSTATE=.TRUE.
ENDIF
RETURN
END SUBROUTINE
C12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901 COLUMN
C      1          2          3          4          5          6          7          8
9 TENS
C              12345   12345   1234567   1234567   1234567   1234567
12345678 FIELD SPAN
C      1          2      3      4 5      6 7      8 9      1011      1213      1415      1617
COMMAS
C              1          2          3          4          5          6          7
FIELD #
C12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901 COLUMN
C0086,02/04/2008,0400,035,*, 0,*, -133,*, 41.4,I, 1.0,*, 73.6,*, 57.0,*,
0.00
C0086,02/04/2008,0500,035,*, 0,*, -133,*, 40.4,I, 1.0,*, 92.6,*, 67.2,*,
0.00
C0086,02/04/2008,0600,035,*, 0,*, -133,*, 41.3,I, 1.0,*, 13.8,*, 35.7,*,
0.00
C0086,02/04/2008,0700,035,*, 0,*, -133,*, 41.4,I, 1.0,*, 24.3,*, 26.6,*,
0.04
C0086,02/04/2008,0800,035,*, 25,*, -115,*, 40.7,I, 1.0,*, 22.2,*, 39.0,*,
0.00
C0086,02/04/2008,0900,035,M,--,M,--,M,--,M,--,M,--,M,--,M,--
C0086,02/04/2008,1000,035,M,--,M,--,M,--,M,--,M,--,M,--,M,--
C0086,02/04/2008,1100,035,M,--,M,--,M,--,M,--,M,--,M,--,M,--
C0086,02/04/2008,1200,035,M,--,M,--,M,--,M,--,M,--,M,--,M,--
C0086,02/04/2008,1300,035,M,--,M,--,M,--,M,--,M,--,M,--,M,--
C0086,02/04/2008,1400,035,M,--,E,--,*, 47.7,I, 0.9,*, 188.0,*, 74.0,*, 0.00
C0086,02/04/2008,1500,035,M,--,*,--,M,--,I, 1.0,*, 277.3,*, 54.5,*, 0.00
C0086,02/04/2008,1600,035,*, 591,E, 259,*, 50.7,I, 1.0,*, 289.2,*, 31.8,*,
0.00
C0086,02/04/2008,1700,035,*, 297,E, 110,*, 50.4,I, 1.0,*, 266.8,*, 26.6,*,
0.00
C0086,02/04/2008,1800,035,*, 17,E, -72,*, 47.7,I, 1.0,*, 274.1,*, 43.6,*,
0.00
C0086,02/04/2008,1900,035,*, 0,E, -84,*, 46.0,I, 1.0,*, 46.1,*, 45.4,*,
0.00
C0086,02/04/2008,2000,035,*, 0,E, -84,*, 45.5,I, 1.0,*, 196.4,*, 65.3,*,
0.00
C0086,02/04/2008,2100,035,*, 0,E, -85,*, 46.4,I, 1.0,*, 137.4,*, 16.8,*,
0.00
C0086,02/04/2008,2200,035,M,--,E,--,M,--,I, 1.0,*, 193.5,*, 37.6,*, 0.00
C0086,02/04/2008,2300,035,M,--,M,--,M,--,I, 1.0,*, 289.3,*, 53.7,*, 0.00
C0086,02/04/2008,2400,035,M,--,M,--,M,--,I, 1.0,*, 270.0,*, 81.0,*, 0.00
C0086,02/05/2008,0100,036,M,--,M,--,M,--,M,--,M,--,M,--,M,--
C0086,02/05/2008,0200,036,M,--,M,--,M,--,M,--,M,--,M,--,M,--

```

PREFORMV2.FOR

```
C      Last change:  BJ   16 Nov 2010   1:51 pm
      PROGRAM PREFORMV2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C 101019 MODIFIED TO MAKE INPUT AND OUTPUT AS "PREFORM INFILE OUTFILE" CALLED PREFORMV2
C ADDED SUBROUTINE IOREADY.FOR, WHICH CHECKS FOR INPUT/OUTPUT FILENAMES ON COMMAND LINE
C AND OPENS UNIT 1 FOR INPUT UNIT 2 FOR OUTPUT BASED ON THOSE NAMES
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C 070921 THIS PROGRAM DESIGNED TO TAKE NEW CIMIS FORMATTED DATA AND CONVERT
C TO OLD FORMAT IN ORDER TO MINIMIZE AMOUNT OF FIDDLING I NEED TO DO WITH CIMPRO
C PROGRAMS TO PROCESS IT.
C
C THIS IS THE OLD FORMAT
C      1      2      3      4      5      6      7      8
9
C123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678901234
C0056,93-01-01,01, 1,*, -0.223,*, -35.636,*, 42.456,*, 6.568,*, 102.800,*,
18.260,*, 0.000
C0056,93-01-01,02, 1,*, -0.304,*, -35.868,*, 42.991,*, 8.289,*, 122.800,*,
19.870,*, 0.000
C0056,93-01-01,03, 1,*, -0.378,*, -35.944,*, 43.345,*, 8.747,*, 150.900,*,
10.910,*, 0.000
C
C THIS IS THE NEW FORMAT BELOW
C      1      2      3      4      5      6      7      8
9
C123456789012345678901234567890123456789012345678901234567890123456789012345678
901
C0039,01/01/2006,0100,001,*, 0,*, -35,*, 42.9,*, 1.4,*, 36.4,*, 22.9,*,
0.00
C0039,01/01/2006,0200,001,*, 0,*, -35,*, 43.4,*, 1.5,*, 35.7,*, 26.8,*,
0.00
C0039,01/01/2006,0300,001,*, 0,*, -36,*, 42.0,*, 1.5,*, 38.0,*, 50.1,*,
0.00
C0039,01/01/2006,0400,001,*, 0,*, -36,*, 41.5,*, 1.6,*, 182.5,*, 50.2,*,
0.00
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCC
      IMPLICIT NONE
      CHARACTER*105 OLDF
      CHARACTER*91 NEWF
C      CHARACTER*40 FIN,FOUT
      INTEGER COMPOS(17),SLPOS(2),DECPOS(5)
      INTEGER LINECOUNT, I,J,K
      INTEGER TREOLD(7),TRENEW(7)
      REAL SR,NR,TEMP,WS,WD,SD,PRECIP

      CHARACTER*35 FIN,FOUT
      LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
      INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS

      DATA TREOLD/21,33,45,57,69,81,93/ !STARTING POSITION FOR 7 QA/QCS OLD RECORD
      DATA TRENEW/25,33,41,51,61,71,81/ !STARTING POSITION FOR 7 QA/QCS NEW RECORD
      DATA COMPOS/5,16,21,25,27,33,35,41,43,51,53,61,63,71,73,81,83/!COMMA LOCATIONS
      DATA SLPOS/8,11/ !SLASH POSITIONS
      DATA DECPOS/49,59,69,79,89/ !DECIMAL POSITIONS

C
      UIN=1
      UOUT=2
C      WRITE(6,12000)(SLPOS(I),I=1,2)
C12000  FORMAT(2I5)
C      WRITE(6,100)
```

```

C100   FORMAT(1X,'ENTER NAME OF INPUT FILE, CONTAINING NEW CIMIS FOR ')
C      READ(5,110)FIN
C110   FORMAT(A40)
C      OPEN(UNIT=1,STATUS='OLD',FILE=FIN)
C      WRITE(6,120)
C120   FORMAT(1X,'ENTER NAME OF OUTPUT FILE, WHICH WILL USE OLD FMT ')
C      READ(5,130)FOUT
C130   FORMAT(A40)
C      OPEN(UNIT=2,STATUS='UNKNOWN',FILE=FOUT)

      CALL IOREADY(INSTATE,OUTSTATE,UIIN,UOUT,FIN,FOUT)
      LINECOUNT=0

C WILL PROCESS LINE BY LINE
1      CONTINUE
      READ(1,200,END=1000)NEWF
200    FORMAT(A91)
      LINECOUNT=LINECOUNT+1
C      DO SOME RUDIMENTARY CHECKING, IT CAN ONLY HELP
      DO I=1,17
        K=COMPOS(I)
        IF(NEWF(K:K).NE.' ')THEN
          WRITE(6,9000)K,K,NEWF(K:K)
9000   FORMAT(1X,'COMMA ERROR ',I5,5X,'NEWF(',I5,')=' ,A1)
          GOTO2000
        ENDIF
      END DO
      DO I=1,2
        K=SLPOS(I)
c      WRITE(6,12500)I,K,NEWF(K:K)
c12500  FORMAT(1X,2I5,5X,A1)
        IF(NEWF(K:K).NE.' / ')THEN
          WRITE(6,9100)K,K,NEWF(SLPOS(I):SLPOS(I))
9100   FORMAT(1X,'SLASH ERROR ',I5,5X,'NEWF(',I5,')=' ,A1)
          GOTO2000
        ENDIF
      END DO
      DO I=1,5
        K=DECPOS(I)
        IF(NEWF(K:K).NE.'. ')THEN
          WRITE(6,9200)K,K,NEWF(DECPOS(I):DECPOS(I))
9200   FORMAT(1X,'DECIMAL ERROR ',I5,'NEWF(',I5,')=' ,A1)
          GOTO2000
        ENDIF
      END DO

      DO I=1,105      !BLANK OUT THE OLD FORMAT LINE
        OLDF(I:I)=' '
      END DO

      !BUILD UP NEW LINE, FIELD BY FIELD

      OLDF(1:5)=NEWF(1:5)
      CALL DATD(NEWF(6:15),OLDF(6:13))
      CALL ICOMM(OLDF,14)
      OLDF(15:16)=NEWF(17:18)
      CALL ICOMM(OLDF,17)
      OLDF(18:20)=NEWF(22:24)
      CALL ICOMM(OLDF,21)

      DO I=1,7      !INSERT THE 7 ',*, ' ( * MAY BE A LETTER CODE TOO)
        CALL ITRE(NEWF,TRENEW(I),OLDF,TREOLD(I))
      END DO

      READ(NEWF(28:32),300,ERR=3000)SR      !SOLAR RADIATION
300    FORMAT(F5.0)
      WRITE(OLDF(24:32),310)SR
310    FORMAT(F9.3)
      READ(NEWF(36:40),320,ERR=3000)NR      !NET RADIATION
320    FORMAT(F5.0)
      WRITE(OLDF(36:44),310)NR

```

```

330      READ(NEWF(44:50),330,ERR=3000)TEMP      !TEMPERATURE
        FORMAT(F7.1)
        WRITE(OLDF(48:56),310)TEMP
        READ(NEWF(54:60),330,ERR=3000)WS      !WIND SPEED
        WRITE(OLDF(60:68),310)WS
        READ(NEWF(64:70),330,ERR=3000)WD      !WIND DIRECTION
        WRITE(OLDF(72:80),310)WD
        READ(NEWF(74:80),330,ERR=3000)SD      !STANDARD DEVIATION WIND DIRECTION
        WRITE(OLDF(84:92),310)SD
340      READ(NEWF(84:91),340,ERR=3000)PRECIP    !PRECIPITATION
        FORMAT(F8.2)
        WRITE(OLDF(96:104),310)PRECIP

        !DONE WITH THAT RECORD, WRITE IT OUT AND MOVE ON TO NEXT RECORD

500      WRITE(2,500)OLDF(1:105)
        FORMAT(A105)
        GOTO1

C          1          2          3          4          5          6          7          8
9
C1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
9012345678901234
C0056,93-01-01,01, 1,*, -0.223,*, -35.636,*, 42.456,*, 6.568,*, 102.800,*,
18.260,*, 0.000
C0056,93-01-01,02, 1,*, -0.304,*, -35.868,*, 42.991,*, 8.289,*, 122.800,*,
19.870,*, 0.000
C0056,93-01-01,03, 1,*, -0.378,*, -35.944,*, 43.345,*, 8.747,*, 150.900,*,
10.910,*, 0.000
C
C THIS IS THE NEW FORMAT BELOW
C          1          2          3          4          5          6          7          8
9
C1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
901
C0039,01/01/2006,0100,001,*, 0,*, -35,*, 42.9,*, 1.4,*, 36.4,*, 22.9,*,
0.00
C0039,01/01/2006,0200,001,*, 0,*, -35,*, 43.4,*, 1.5,*, 35.7,*, 26.8,*,
0.00
C0039,01/01/2006,0300,001,*, 0,*, -36,*, 42.0,*, 1.5,*, 38.0,*, 50.1,*,
0.00
C0039,01/01/2006,0400,001,*, 0,*, -36,*, 41.5,*, 1.6,*, 182.5,*, 50.2,*,
0.00

1000     CLOSE(1)
        CLOSE(2)
        WRITE(6,1100)LINECOUNT
1100     FORMAT(1X,'EOF ON ',I5,' PROGRAM TERMINATED NORMALLY')
        STOP
2000     WRITE(6,2100)LINECOUNT,NEWF
        WRITE(2,2100)LINECOUNT,NEWF
2100     FORMAT(1X,'WRONG COMMA,SLASH,OR DECIMAL ',/1X,'LINE#= ',I5,
1         /1X,A91)
        STOP

3000     CONTINUE !REACHES HERE IF READ ERROR IN ONE OF THE FIELDS
        WRITE(6,3100)LINECOUNT,NEWF,OLDF
        WRITE(2,3100)LINECOUNT,NEWF,OLDF
3100     FORMAT(1X,'READ ERROR IN LINE ',I5,/1X,A91/1X,A105)
        STOP

END PROGRAM

SUBROUTINE ITRE(NEWF,K,OLDF,L)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C INSERT 3 CHARACTER STRING (GENERALLY WILL BE ,*,)
C FROM NEWF STARTING AT K TO OLDF STARTING AT L
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
IMPLICIT NONE

```



```

        INTEGER K,L
        CHARACTER*91 NEWF
        CHARACTER*105 OLDF
        OLDF(L:L+2)=NEWF(K:K+2)
        RETURN
        END SUBROUTINE

        SUBROUTINE ICOMM(OLDF,N)
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C INSERT COMMA AT POSITION N
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
        IMPLICIT NONE
        INTEGER N
        CHARACTER*91 OLDF
        OLDF(N:N)=' ,'
        RETURN
        END SUBROUTINE

        SUBROUTINE DATD(N,O)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE DATD CONVERTS 01/01/2006 (MM/DD/YEAR) TO 06-01-01 (YY-MM-DD)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        CHARACTER*10 N
        CHARACTER*8 O
        O(1:2)=N(9:10)
        O(3:3)='- '
        O(4:5)=N(1:2)
        O(6:6)='- '
        O(7:8)=N(4:5)
        RETURN
        END SUBROUTINE
C      Last change:  BJ  19 Oct 2010    4:31 pm
        SUBROUTINE IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CALLING PROGRAM NEEDS TO DECLARE THE FOLLOWING
C-----
C   CHARACTER*35 FIN,FOUT
C   LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
C   INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
C   UIN=1
C   UOUT=2
C-----
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        INTEGER UIN,UOUT
        CHARACTER*35 FIN,FOUT
        LOGICAL INSTATE,OUTSTATE,INSTATUS
        CHARACTER*70 CLINE

        CALL GETCL(CLINE)
        CALL IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)

        IF(INSTATE.AND.OUTSTATE)THEN
C           WRITE(6,500)FIN,FOUT,K1,K2
C500      FORMAT(1X,'IOREADY: INPUT,OUTPUT ',/1X,A35,/1X,A35,
C          * /1X,'K1,K2 ',2I3)
          CONTINUE !STUFF ABOVE FOR DEBUGGING
        ELSEIF (.NOT.INSTATE)THEN
          WRITE(6,505)FIN,K1
C505      FORMAT(1X,'IOREADY: INPUT FILENAME ERROR ',A35,1X,I3)
          STOP
        ELSEIF (.NOT.OUTSTATE)THEN
          WRITE(6,510)FOUT,K2
C510      FORMAT(1X,'IOREADY: OUTPUT FILENAME ERROR ',A35,1X,I3)
          STOP
        ENDIF

```

```

!CHECK TO MAKE SURE THE INPUT FILE EXISTS
INQUIRE(FILE=FIN,EXIST=INSTATUS)
IF(.NOT.INSTATUS)THEN
  WRITE(6,255)FIN
255  *   FORMAT(1X,'IOREADY: INPUT FILE NOT EXIST',
        /1X,A35, /1X,'PROGRAM STOPPING ')
      STOP
      ENDIF
      OPEN(UNIT=UIN,STATUS='OLD',FILE=FIN,ERR=1000)
      OPEN(UNIT=UOUT,STATUS='UNKNOWN',FILE=FOUT,ERR=2000)
      RETURN

!ERRORS ON OPENING GO BELOW
1000  CONTINUE
      WRITE(6,1100)FIN
1100  FORMAT(1X,'ERROR TRYING TO OPEN IN FILE ',/1X,A35)
      STOP
2000  CONTINUE
      WRITE(6,2100)FOUT
2100  FORMAT(1X,'ERROR TRYING TO OPEN OUT FILE ',/1X,A35)
      STOP
      END SUBROUTINE

SUBROUTINE IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)
IMPLICIT NONE
CHARACTER*35 FIN,FOUT
CHARACTER*70 CLINE
INTEGER K1,K2,IBLANK,K,I,L
LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
!K1, K2 ARE LENGTHS OF THE FILENAMES

DO I=1,35
  FIN(I:I)=' '
  FOUT(I:I)=' '
END DO
!ASSUME CLINE HAS 1 OR MORE BLANK SPACES WHICH DIVIDE THE TWO FILENAMES
!ASSUME THAT INPUT FILE IS THE FIRST FILE, AND OUTPUTFILE IS THE SECOND ONE
K=LEN_TRIM(CLINE) !FIND OUT TOTAL STRING LENGTH
!LOOK FOR FIRST BLANK
IBLANK=0
DO I=1,K
  IF(CLINE(I:I).EQ.' ')THEN
    IBLANK=I
    GOTO100
  ELSE
    CONTINUE
  ENDIF
END DO
100  IF (IBLANK.EQ.0) THEN
      INSTATE=.FALSE.
      OUTSTATE=.FALSE.
      K1=0
      K2=0
      WRITE(6,200)CLINE
200  FORMAT(1X,'COMMAND LINE LACKING 2 FILENAMES ',/1X,A70)
      RETURN
    ELSEIF (IBLANK.GE.70)THEN
      WRITE(6,250)CLINE
250  FORMAT(1X,'FILENAME TOO LONG ',/1X,A70)
      K1=IBLANK
      K2=0
      INSTATE=.TRUE.
      OUTSTATE=.FALSE.
    ELSE
      FIN(1:IBLANK-1)=CLINE(1:IBLANK-1)
      L=K-(IBLANK-1)
      FOUT(1:L)=CLINE(IBLANK:K)
      FOUT(1:L)=ADJUSTL(FOUT(1:L))
      K2=LEN_TRIM(FOUT(1:L))
      IF(K2.EQ.0)THEN
        INSTATE=.TRUE.

```

```
      OUTSTATE=.FALSE.  
      RETURN  
    ENDIF  
    K1=IBLANK-1  
    INSTATE=.TRUE.  
    OUTSTATE=.TRUE.  
  ENDIF  
  RETURN  
END SUBROUTINE
```

CIMPRO1V4.FOR

```
C      Last change:  BJ   17 Nov 2010   4:47 pm
C      PROGRAM CIMPRO1v4
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C BRJ101115 ADDED 'N' QC FLAG TO VALIDA SUBROUTINE BECAUSE NEEDED TO EXCLUDE
C CASE WHERE NET RADIATION (OR ANY VALUE) IS NOT CALCULATED BY STATION
C SO THIS IS NOW CIMPPRO1 VERSION 4.
C BRJ101019 AM MODIFYING CIMPRO1V2 AS FOLLOWS:
C   1. CHANGING THE POLICY TO INCLUDE "R" AS ACCEPTABLE, BASED ON REVIEW OF
C     R VALUES, HAVE DECIDED TO INCLUDE THESE AS ACCEPTABLE
C   2. ELIMINATING SOME OF THE TEXT AT BEGINNING OF PROGRAM, WHICH WILL BE INCLUDED IN
C     DOCUMENTATION FOR RUNNING THE PROGRAM (SO DON'T HAVE TO HIT RETURN A BUNCH OF
C     TIMES)
C   3. MAKING THE INPUT/OUTPUT UNIFORM WITH THE OTHER 3 PROGRAMS USED TO PROCESS CIMIS
C
C     NAMEDLY, CIMPRO1V3 INFILE OUTFILE
C
C     SINCE CIMPRO1V2 PRODUCES 2 OUTPUT FILES (A *.SUM FILE WHICH IS A SUMMARY FILE
C     SHOWING
C       THE BAD DAYS AND RECKONING AND A *.DET FILE WHICH IS THE PROCESSED MET DATA,
C       THE SECOND FILE ON THE COMMAND LINE WILL BE ASSUMED TO BE THE BJ1 FILE AND THE
C       PROGRAM WILL LOOK FOR A '.' AND REPLACE THE BJ1 WITH A SUM AND PROCEED
C     ACCORDINGLY
C
C       THE UNITS WILL BE (IN ORDER TO BE CONSISTENT WITH THE EARLIER CIMPRO1 VERSION)
C       7=INPUT FROM OPEN DATA FILE
C       2=OUTPUT TO THE SUM FILE
C       8=OUTPUT TO THE BJ1 FILE
C     THIS FEATURE WILL ALSO FACILITATE BATCH PROCESSING WITH ARBITRARY NAMES., THIS
C     FEATURE IS FACILITATED BY THE ADDITION OF A SUBROUTINE CALLED "IOREADY.FOR"
C     WHICH OPENS UNIT1 FOR READING AND UNIT2 FOR OUTPUT AND DOES SOME CHECKING
C     TO MAKE SURE THAT THE INPUT FILE EXISTS AND THAT BOTH FILES HAVE REASONABLE NAMES
C     I COMMENTED OUT THE 'OPF1 LOGICAL FUNCTION WHICH OPENED UNITS 1 AND 2 BASED ON THE
C     FILENAME IN CHARACTER STRING ASCII, THIS WILL BE REPLACED BY IOREADY.
C
C     IN ADDITION TO IOREADY, I ALSO WROTE SUPPLEMENTAL SUBROUTINE TO OPEN THE SUM FILE
C     THE BJ1 FILE WILL BE GIVEN ON THE COMMAND LINE (CIMPRO1V3 INFILE OUTFILE) AS
C     OUTFILE
C     THE IOSUPP SUBROUTINE TAKES THE OUTFILE AND ADDS OR CHANGES THE EXTENSION TO
C     '.SUM'
C     AND OPENS IT AS UNIT 2
C
C     SUMMARY OF CHANGES,
C     ADDED 2 SUBROUTINES IOREADY.FOR AND IOSUPP.FOR, COMMENTED OUT OPF1 AND ADDED A
C     FEW LINES
C     TO MAIN PROGRAM TO INSERT THE VERSION OF CIMPRO1V3 INTO THE SUM FILE.
CCCCCCCC
C
C
C BRJ070926 DECIDED TO INCLUDE THE FOLLOWING FLAGS AS PROBLEMS: I,M,S, AND R
C USER CAN DECIDE IF WANT TO INCLUDE R OR NOT
C I WOULD RECOMMEND RUNNING THE FIRST WITH INCLUDING 'R', AND THEN REVIEWING THE RS AND
C FIGURING OUT WHAT TO DO FROM THERE.
C
C BRJ070925 THIS IS A REVISED VERSION OF CIMPRO1. THE PURPOSE OF THE REVISION IS TO
C TAKE INTO ACCOUNT SOMEWHAT DIFFERENT MISSING VALUE CONVENTIONS AND CODES PRESENT IN
C CIMIS DATA SINCE 1995. WHEN I LAST WORKED ON THIS STUFF, CIMIS WAS STILL IN TRANSITION
C AND HADN'T SHIFT OVER YET. I ORIGINALLY INTENDED TO MODIFY CIMPRO1 TO ALSO TAKE INTO
C ACCOUNT THE DIFFERENT FORMATTING. HOWEVER, IT DETERMINED THAT IT WILL BE EASIER TO
C CREATE A PROGRAM (PREFORM.FOR), WHICH REFORMATS THE NEW DATA INTO THE OLD FORMAT.
C THEREFORE, CIMPRO1V2 WILL ASSUME THE OLD FORMAT AND BY OLD FORMAT I MEAN THE FIELD/
C COLUMN DEFINITIONS. THE ORDER OF THE COLUMNS IS STILL THE SAME, BUT THE LENGTH OF
C COLUMNS VARIES AND ARE GENERALLY SHORTER THAN THEY USED TO BE (USED TO BE F9). I HAVE
C A COMPLETE DEFINITION OF THE NEW COLUMNS AND FIELDS IN A FILE CALLED NEW-CIMIS-
C FIELDS.XLS
C
C BUT I HAVE SUCCESSFULLY WRITTEN PREFORM AND TESTED IT AND IT'S OK. SO, FORMAT IS THE
C SAME
```

```

C NOW AS OLD. BUT GO THROUGH CODE AND ADJUST QC FLAGS DETECTION AND MISSING VALUE WHERE
C NEEDED (CHECK LOOK4D, FOR EXAMPLE).
C
C NEW SEVERE CODES ARE I,M,S, OLD SEVERE CODES WERE I,S. I NOTE THAT THE ONLY USE OF
C "I" IS IN THE PARLIER 2006 DATA WAS THE WIND SPEED WHEN THE SPEED WAS 1MPH, UNDOUBTEDLY
C BECAUSE
C THE MINIMUM VALUE REPORTED BY THE INSTRUMENT IS 1.0 MPH, EVEN IF THE TRUE WIND SPEED IS
C 0. THIS DOES NOT POSE A PROBLEM AT THIS TIME BECAUSE THESE VALUES WERE GENERALLY
C LUMPED
C INTO CALMS.
C
C ALSO THEY USE 2 DASHES OCCASIONALLY WHERE DATA IS MISSING. IN MY CASE, I WON'T HAVE
C ANY
C MISSING DATA BECAUSE I'VE HAD TO INTERPOLATE TO GET AT LEAST A NUMBER IN EVERY PLACE
C IN ORDER TO RUN PREFORM. PREFORM USES FORTRAN NUMERIC READS TO TRANSFER THE
C INFORMATION,
C (I.E. NOT JUST CHARACTER MANIPULATION). THESE READS WOULD BOMB OUT IF THERE WERE
C ANYTHING
C OTHER THAN NUMERIC DATA IN THE FIELD.
C
C HERE IS THE DATA FORMAT AFTER RUNNING PREFORM
C      1      2      3      4      5      6      7      8
C      10
C123456789012345678901234567890123456789012345678901234567890123456789012345678
90123456789012345
C0039,06-01-01,01,001,*, 0.000,*, -35.000,*, 42.900,*, 1.400,*, 36.400,*,
22.900,*, 0.000
C0039,06-01-01,02,001,*, 0.000,*, -35.000,*, 43.400,*, 1.500,*, 35.700,*,
26.800,*, 0.000
C0039,06-01-01,03,001,*, 0.000,*, -36.000,*, 42.000,*, 1.500,*, 38.000,*,
50.100,*, 0.000
C
C I MANUALLY CHECKED THIS FORMAT AGAINST THE MAIN READ STATEMENT AND IT APPEARS TO BE OK
C PURPOSE IS TO CHECK CIMIS DATA FOR CONSISTENCY, MISSING VALUES
C INTERPOLATE WHERE POSSIBLE, AND FLAG UNUSABLE DAYS IN PREPARATION
C FOR RUNNING CIMPRO2
C
C CIMPRO1 OUTPUTS PROBLEM RECORDS, AND COUNTS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
INTEGER STATN, YEAR, MONTH, JULDAY, HOUR
INTEGER TOTDAY, GODDAZ, BADDZ
REAL NETRAD, AIRT, WVEL, WDIR, SIGMAT
CHARACTER NETRAQ, AIRTQ, WVELQ, WDIRQ, SIGMAQ
CHARACTER*10 ERRMESS
CHARACTER*105 LINE(24)
LOGICAL FREC,OPF1
CHARACTER*11 ASC1
INTEGER OSTATN, OYEAR, OMONTH, OJULD, OHOUR, OSTAB, RECORD
DIMENSION STATN(24), YEAR(24), MONTH(24), JULDAY(24), HOUR(24)
DIMENSION NETRAD(24), AIRT(24), WVEL(24), WDIR(24), SIGMAT(24)
DIMENSION NETRAQ(24), AIRTQ(24), WVELQ(24), WDIRQ(24), SIGMAQ(24)
LOGICAL RADREJ,AIRREJ,WVREJ,WDREJ,STREJ
LOGICAL INCLR !FLAG FOR WHETHER OR NOT TO INCLUDE 'R' AS BAD VALUE BRJ070927
COMMON /REJCRI/RC,INCLR !BRJ070927
INTEGER RC !REJECTION CRITERIA
LOGICAL ASK

!BRJ101020 STUFF BELOW ADDED FOR IOREADY ROUTINE

CHARACTER*35 FIN,FOUT
LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS

!BRJ101021 STUFF BELOW ADDED FOR IOSUPP ROUTINE

LOGICAL OUT2STATE !SUCCESS = TRUE, FAILURE = FALSE
INTEGER U2OUT !OUTPUT UNIT, WHICH IN CALLING PROGRAM WILL PROBABLY BE 2
C CHARACTER*35 FOUT !OUTPUT FILENAME ALREADY OPENED (BJ1 FILENAME)
CHARACTER*39 F2OUT !OTUPUT FILE CREATED IN IOSUPP, WITH .SUM EXTENSION

```

```

      UIN=7          !INPUT BRJ101020 LOOKS LIKE THE READ IS FROM UNIT 7
      UOUT=8        !THIS IS THE BJ1 OUTPUT FILE AND IS USED BY IOSUPP TO CREATE U2OUT
FILENAME
      U2OUT=2      !THIS IS THE SUM OUTPUT FILE

      RC=4 !NOTE, IF THERE ARE MORE THAN 'RC' RECORDS BAD IN A DAY, THEN WHOLE DAY IS
REJECTED
C      INCLR=.TRUE. !INCLUDE IT UNTIL TOLD OTHERWISE !BRJ101020 INCLR SET TO FALSE
BELOW, SO THAT
C      R FLAGGED DATA WILL BE CONSIDERED LEGIT

C      CALL INTRO() !BRIEF INTRODUCTORY MATERIAL !BRJ101020 NO PUT THIS IN
DOCUMENTAITON NOW

!BRJ101020 THIS NEXT SECTION COMMENTED OUT, EXCEPT SET INCLR TO FALSE TO INCLUDE THE R
FLAGGED VALUES AS LEGIT DATA
C      IF(ASK('DO YOU WANT "R" FLAG AS BAD VALUE ',34))THEN          !BRJ070927
C      INCLR=.TRUE. !TRUE MEANS R WILL BE INCLUDED WITH I,M,S AS BAD VALUES
C      ELSE
C      INCLR=.FALSE. !THIS MEANS THAT R FLAGGED DATA WILL BE CONSIDERED LEGIT
BRJ101020
C      ENDIF
C
C
C BRJ101020 NOPE, COMMENT OUT THIS FILE BUSINESS AND REPLACE WITH "IOREADY" SUBROUTINE
C OPEN FILE CONTAINING LIST OF FILES TO PROCESS, EACH FILE ON
C SEPARATE LINE, 11 CHARACTERS IN FILENAME

C      OPEN(UNIT=13,STATUS='OLD',FILE='ASCLIST.DAT')
C99221 READ(13,1333,END=99771)ASC1
C1333  FORMAT(A11)
      CALL IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)
      IF(INSTATE.AND.OUTSTATE)THEN
      CALL IOSUPP(OUT2STATE,U2OUT,FOUT,F2OUT)
      ENDIF
      IF(.NOT.(INSTATE.AND.OUTSTATE.AND.OUT2STATE))THEN
      WRITE(6,7812)
7812  FORMAT(1X,'CIMPRO1: FILE OPEN PROBLEM...STOPPING')
      STOP
      ENDIF
C      IF(OPF1(ASC1))THEN !BRJ101020 NOPE, DONT USE THIS ANYMORE
      IF(INSTATE.AND.OUTSTATE.AND.OUT2STATE)THEN !BRJ101020

      !SINCE OPF1 COMMENTED OUT, NEED TO WRITE CIMPRO1 VERSION TO UNIT 2 NOW HERE
      WRITE(2,7912)FIN
7912  FORMAT(1X,'CIMPRO1V4 ...',/1X,'INPUT FILE: ',A35)
      WRITE(2,7913)FOUT
7913  FORMAT(1X,'PROCESSED OUTPUT FILE ',A35) !IE *.DET FILE
      WRITE(2,7914)F2OUT
7914  FORMAT(1X,'SUMMARY FILE ',A39) !IE *.SUM FILE
C
C      INITIALIZE

      RECORD = 0
      OSTATN = 0 !FIRST TIME IS 0, AFTER IS PREV HOUR'S VALUE
      OYEAR = 0
      OMONTH = 0
      OJULD = 0
      OSTAB = 0 !USED IN CORRECTING STABILITY CLASSES LATER ON
      OHOUR = 0

      WRITE (*,*)
      WRITE (*,*) 'ONLY DAYS WITH BAD OR MISSING VALUES ARE LISTED.'
      WRITE (2,*) 'ONLY DAYS WITH BAD OR MISSING VALUES ARE LISTED.'
      WRITE (*,*)
      1'AN * ON LEFT INDICATES THAT ALL 24H ARE BAD FOR THAT SENSOR.'
      WRITE (*,*)
      1'AN * ON RIGHT INDICATES THAT DAY WILL BE MARKED UNUSABLE.'
      WRITE (*,*)

```

```

WRITE (*,*)
1' RECORD DAY SENSOR      HOURLY FLAG : GOOD/BAD'
WRITE (2,*)
1'AN * ON LEFT INDICATES THAT ALL 24H ARE BAD FOR THAT SENSOR.'
WRITE(2,*)
1'AN * ON RIGHT INDICATES THAT DAY WILL BE MARKED UNUSABLE.'
WRITE (2,*)
WRITE (2,*)
1' RECORD DAY SENSOR      HOURLY FLAG : GOOD/BAD'

WRITE (*,*)
1' _____
2' _____'
WRITE (2,*)
1' _____
2' _____'
WRITE (*,*)
c WRITE (2,*)
c 1' RECORD DAY SENSOR      HOURLY FLAG : GOOD/BAD'
c
c WRITE (2,*)
c 1' _____
c 2' _____'
c WRITE (2,*)
C
C LOOP OVER CALCULATIONS - DAY BY DAY

C
C SOME COUNTERS TO KEEP TRACK OF DAYS REJECTED
GODDAZ=0
BADDAZ=0
FREC=.TRUE.
10 CONTINUE

DO 20 I=1,24
2879 READ(7,3991,END=997)LINE(I)(1:104) !READ IN LINE AS CHARACTERS, THEN PROCESS
LATER
3991 FORMAT(A104)
c WRITE(6,4422)I,LINE(I)
c4422 FORMAT(1X,'LINE # ',I5,/1X,A104)

C LOOK4D MUST LOOK FOR BOTH '*****' AND ' --'
C BRJ070925 CHANGED SUBROUTINE TO LOOK4E, AND LOOKS FOR '**' AND '--'
C WON'T FIND ANY IN PARLIER DATA, SINCE I ALREADY TOOK CARE OF MANUALLY

CALL LOOK4E(LINE(I),RECORD-1,2)
FREC=.FALSE. !NOTE: THIS CAUSES PERMANENT SKIPPING OF 'IF' IMMEDIATELY FOLLOWING,
MUST BE AN ANCIENT CHANGE
IF(FREC)THEN
1724 WRITE(0,1643)LINE(I)(1:80)
1643 FORMAT(1X,'DO YOU WISH TO INCLUDE 1ST LINE IN PROCESS',
1 /1X,A80,
1 /1X,'0=NO, 1=YES ')
READ(0,1688)LIOP
1688 FORMAT(I1)
IF(LIOP.EQ.0)THEN
FREC=.FALSE.
GOTO2879
ELSEIF(LIOP.EQ.1)THEN
FREC=.FALSE.
CONTINUE
ELSE
WRITE(0,5557)
5557 FORMAT(1X,'ANSWER WITH 0 OR 1 ONLY ')
GOTO1724
ENDIF
ENDIF

C
C THIS READS ONE LINE OF DATA, WHICH SHOULD HAVE FORMAT AS FOLLOWS:

```

```

C BRJ070925 I CHECKED THIS FORMAT, IT'S OK, AND NOTE THAT IT SKIPS SOLAR RADIATION, WHICH
IS THE
C FIRST VARIABLE IN MY DOWNLOADS, BUT NEEDS SOLAR RADIATION TO GET THE FORMATTING OF THE
RECORDS
C RIGHT
C

```

```

      READ (LINE(I),9101, END=997, ERR=998) STATN(I), YEAR(I),
2      MONTH(I),
3      HOUR(I), JULDAY(I), NETRAQ(I), NETRAD(I), AIRTQ(I), AIRT(I),
4      WVVELQ(I), WVVEL(I),WDIRQ(I), WDIR(I), SIGMAQ(I), SIGMAT(I)
9101  FORMAT(I4, T6, I2, T9, I2, T15, I2, T18, I3, T34, A1, T36, F9.0,
1      T46, A1, T48, F9.0, T58, A1, T60, F9.0, T70, A1, T72, F9.0,
2      T82,A1, T84,F9.0)

```

```

      RECORD = RECORD + 1

```

```

C
C CHECK DATA FOR INTERNAL CONSISTENCY, DOES IT SKIP ANYTHING ?

```

```

      CALL ERRCHK(A,STATN(I), YEAR(I), MONTH(I), JULDAY(I), HOUR(I),
1      OSTATN, OYEAR, OMONTH, OJULD, OHOUR, RECORD)

```

```

20  CONTINUE      !END OF READ FOR THIS DAY
c      WRITE(6,15000)year(1),month(1),julday(1)
c15000  FORMAT(1x,'now checking qc for yy,mo,jday ',3(i4,', '))
      ERRMESS = 'NETRAD'
      CALL CHECKQA(NETRAD, NETRAQ, RECORD, ERRMESS, JULDAY(1),RADREJ )
      ERRMESS = 'AIRTEMP'
      CALL CHECKQA(AIRT, AIRTQ, RECORD, ERRMESS, JULDAY(1),AIRREJ )
      ERRMESS = 'WINDVEL'
      CALL CHECKQA(WVVEL, WVVELQ, RECORD, ERRMESS, JULDAY(1),WVREJ )
      ERRMESS = 'WINDDIR'
      CALL CHECKQA(WDIR, WDIRQ, RECORD, ERRMESS, JULDAY(1),WDREJ )
      ERRMESS = 'SIGMATHETA'
      CALL CHECKQA(SIGMAT, SIGMAQ, RECORD, ERRMESS, JULDAY(1),STREJ )

```

```

C
C FIGURE OUT IF THIS DAY GETS REJECTED
C FOR NOW , CRITERIA IS REJECTION OF ANY OF THE 5 VARIABLES ABOVE
      IF(RADREJ.OR.AIRREJ.OR.WVREJ.OR.WDREJ.OR.STREJ)THEN

```

```

      BADDAZ=BADDAZ+1
      DO 7027 K=1,24
7027  LINE(K)(105:105)='#' !PUT # INTO POSITION 105, INDICATING THIS DAY NOT 2B USED
      ELSE
      DO 7291 K=1,24
7291  LINE(K)(105:105)=' '
      GODDAZ=GODDAZ+1
      ENDIF

```

```

      DO 4778 K=1,24
4778  WRITE(8,4779)LINE(K)
4779  FORMAT(A105)
5      GOTO10 !GO BACK UP AND GET THE NEXT DAY

```

```

C
C WRITE OUT NUMBER OF RECORDS, AND NUMBER OF GOOD AND BAD DAYS

```

```

997      TOTDAY=RECORD/24
      WRITE(0,1723)RC,RECORD,TOTDAY,GODDAZ,BADDAZ,GODDAZ+BADDAZ
      WRITE(2,1723)RC,RECORD,TOTDAY,GODDAZ,BADDAZ,GODDAZ+BADDAZ
1723  FORMAT(1X,'CRITERIA FOR REJECTING DAYS: WHEN >',I2,
1      ' RECORDS ARE BAD FOR AT LEAST 1 VARIABLE ',
2/1X,I8,' RECORDS EXAMINED WHICH IS ',I8,' DAYS',
1/1X,'GOOD DAYS= ',I6,', BAD DAYS= ',I6,' , SUM OF BOTH= ',I6)
      IF(INCLR)THEN !BRJ070927
      WRITE(2,1993)
      WRITE(6,1993)
1993  FORMAT(1X,'R WAS INTERPRETED AS BAD DATA IN THIS RUN ')
      ELSE
      WRITE(2,1994)

```



```

        WRITE(6,1994)
1994   FORMAT(1X,'R WAS NOT CONSIDERED AS BAD DATA IN THIS RUN')
        ENDIF

        GOTO55270

C
C   ERROR OCCURED IN I/O - PRINT ERROR MESSAGE AND QUIT

998   WRITE (*,*) 'I/O ERROR WHILE ATTEMPTING TO READ MET DATA'
        WRITE (*,*) 'ERROR WAS IN SAME DAY AS RECORD', RECORD
        WRITE (*,*) 'PROGRAM ENDED DUE TO ERROR.'
        WRITE (2,*) 'I/O ERROR WHILE ATTEMPTING TO READ MET DATA'
        WRITE (2,*) 'ERROR WAS IN SAME DAY AS RECORD', RECORD
        WRITE (2,*) 'PROGRAM ENDED DUE TO ERROR.'

        ENDIF
55270   CLOSE(2)
        CLOSE(7)
        CLOSE(8)
C   GOTO99221 !BRJ101020 IN PREVIOUS VERSION THIS GOES TO NEXT LINE IN ASCLIST, NOW
NOT NEEDED
99771   STOP
        END PROGRAM

        SUBROUTINE CHECKQA (ARRAY, ARRAYQ, RECORD, ERMESS, JULDAY, REJ)
CCCCCCCCCCCCCCCCCCCC BRJ070925 CCCCCCCCCCCCCCCCCCCCCCCCCC
C
C I HAVE CHANGED THE QAQC STUFF SOME. (1) CIMIS CHANGED THEIR FLAGS SOME
C (2) I'M LOOSENING UP THE RESTRICTIONS SOME (3) USER DETERMINES IF
C 'R' IS BAD DATA OR NOT INCLR IS FLAG TRUE --> YES 'R' IS BAD
C 'R' IN THE LIST OF FLAGS TO WORRY ABOUT
C
C CURRENT HOURLY SENSOR FLAG SUMMARY (FROM CIMIS WEBSITE 9/26/07)
C
C SEVERE FLAGS
C 'I' = DATA ALUE HAS NO MEANING/IGNORE (NOTE: FOR 2006 PARLIER, THIS WAS ONLY USED
C                                     FOR WIND SPEEDS AT 1.0MPH PROBABLY BECAUSE
C                                     THOSE DEVICES DON'T REPORT ANYTHING LESS.
C                                     I WILL HANDLE THIS CASE BY IGNORING IT AND
C                                     USING THE 1MPH - THESE GENERALLY CLASSSED AS
C                                     CALMS ANYWAY)
C 'M' = DATA VALUE MISSING
C 'S' = SENSOR OUT OF SERVICE OR DATA OUT OF SENSOR THRESHOLD
C
C INFORMATIVE FLAGS
C 'A' = HISTORICAL AVG
C 'E', 'T' = HISTORICAL AVG USED TO CALCULATE VALUE (NOTE: I HAVE ALSO USED AN 'E' TO
INDICATE
C                                     THAT I INTERPOLATED A VALUE)
C 'N' = DATA VALUE NOT COLLECTED BY THIS STATION
C 'P' = QUALITY TEST PENDING
C 'Q' = QC COULD NOT BE PERFORMED
C 'R' = DATA FAR OUT OF HISTORICAL LIMITS (NOTE: THIS FLAG IS OPTIONAL, USER CAN
C                                     DECIDE WHETHER IT'S BAD VALUE OR NOT)
C 'Y' = DATA MODERATELY OUT OF HISTORICAL LIMITS
C
C MY INTENTION IS TO USE I M S FLAGS AND CALL THOSE DATA INVALID, THE REST WILL BE VALID.
C
C END OF BRJ070925 COMMENTS CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C BEGINNING OF COMMENTS FROM THE OLDEN DAYS
C
CCCCC CHECKQA LOOKS FOR QUALITY FLAGS WHICH ARE NOT '*' OR 'F'. IT REPLACES
C THESE BAD VALUES WITH THE AVERAGE OF THE TWO ADJOINING VALUES. A BAD
C VALUE IN HOUR 1 IS REPLACED WITH THE FIRST VALID VALUE, AND A BAD VALUE
C IN HOUR 24 IS REPLACED BY THE LAST VALID VALUE.
C
C '*' = NORMAL' ' = ALSO NORMAL
C 'C' = NOT COLLECTED 'S' = SENSOR OUT OF SERVICE
C 'M' = NOT AVAILABLE 'R' = OUT OF RANGE SEVERE
C 'F' = ESTIMATED 'Y' = OUT OF RANGE

```

```

C
C     IS THE CHARACTER A VALID ONE ?
C
C LOGICAL REJ TRUE IF THIS DAY OF DATA MUST BE REJECTED
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      REAL ARRAY
      LOGICAL INCLR                               !BRJ070927
      COMMON /REJCRI/RC,INCLR                    !BRJ070927
      INTEGER RC
      CHARACTER ARRAYQ, STAR,STAR2
      CHARACTER*10 ERRMESS
      CHARACTER ISGOOD
      LOGICAL ANYERR, ALLERR
      LOGICAL REJ

      LOGICAL VALIDA
      INTEGER RECORD, JULDAY

      DIMENSION ARRAY(24), ARRAYQ(24), ISGOOD(24)

      ANYERR = .FALSE.
      ALLERR = .TRUE.

C
C KEEP TRACK OF THE NUMBER OF ERRORS FOR REJECTION CRITERIA
      NUMERR=0
      DO 30 I=1,24
      IF (VALIDA(ARRAYQ(I),ERRMESS)) THEN !BRJ070926 ADDED ERRMESS ARGUMENT
      ISGOOD(I)= 'G'
      ALLERR = .FALSE.
      ELSE
c      WRITE(6,15000)i,arrayq(i),errmess
c15000   FORMAT(1x,'CHECKQA: invalid data hh,data,errmess ',i2,',',a1,
c      1   ', ',a10)
      ISGOOD(I)= 'B'
      ANYERR = .TRUE.
      NUMERR=NUMERR+1
      END IF
30      CONTINUE
      IF (ALLERR) THEN
      STAR = '*'
      ELSE
      STAR = ' '
      END IF
C
C REJECTION CRITERIA

      IF(NUMERR.GT.RC)THEN
      REJ=.TRUE.
      STAR2='*'
      ELSE
      REJ=.FALSE.
      STAR2=' '
      ENDIF
C      IF (ANYERR) THEN
      IF(NUMERR.GT.1)THEN
      WRITE (*,301) STAR, RECORD, JULDAY, ERRMESS, ISGOOD,NUMERR,STAR2
      WRITE (2,301) STAR, RECORD, JULDAY, ERRMESS, ISGOOD,NUMERR,STAR2
      END IF
301   FORMAT (' ', A1, I6, I4, A11, 24A2,1X,I2,A1)
      END

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      LOGICAL FUNCTION VALIDA (QFLAG,ERRMESS)
      !BRJ101115 ALSO LOOKED AT POSSIBLY INCLUDING 'Q' FLAG, THIS IS TRICKY, USUALLY
OCCURS
      !FROM EXAMPLES I'VE SEEN WHEN PROBLEM WITH SOLAR RADIATION, AND USUALLY THERE ARE
A BUNCH

```

```

!OF M AND S FLAGS ON THE SAME DAY, SO IT GETS EXCLUDED
!SO, FOR NOW, I WON'T WORRY ABOUT Q BECAUSE IT ONLY SEEMS TO OCCUR ON NET
RADIATION
!AND MANY TIMES THE NET RADIATION IS CORRECT IN THE SENSE OF NIGHT AND DAY
!THOUGH I DID SEE SOME EXAMPLES WHERE IT WASN'T, SO WILL PUT BURDEN ON USER TO
!WORRY ABOUT, AS WITH THE R FLAG
CHARACTER*1 QFLAG
!BRJ101115 ADDED CHECKING FOR "N", NOT CALCULATED
CHARACTER*10 ERRMESS !BRJ070926 ADDED ERRMESS TO DEAL WITH WINDSPEED=1.0MPH
SITUATION
LOGICAL INCLR !BRJ070927
COMMON /REJCRI/RC,INCLR !BRJ070927
VALIDA=.TRUE. !INNOCENT UNTIL PROVEN GUILTY THIS TIME
C IF((QFLAG.EQ.' ').OR.(QFLAG.EQ.'*').OR.(QFLAG.EQ.'F'))THEN
C VALID = .TRUE.
C ELSE
C VALID = .FALSE.
C END IF
IF(QFLAG.EQ.'M'.OR.QFLAG.EQ.'S'.OR.QFLAG.EQ.'N')THEN !INCLUDING M,S,N HERE
VALIDA=.FALSE.
RETURN
ELSEIF(QFLAG.EQ.'I')THEN
IF (ERRMESS.EQ.'WINDVEL') THEN !IF THIS IS WINDSPEED, THEN ASSUME VALID TRUE DUE
TO 1MPH SPEED.
VALIDA=.TRUE.
RETURN
ELSE
VALIDA=.FALSE.
RETURN
ENDIF
ELSEIF(QFLAG.EQ.'R')THEN !BRJ070927
IF (INCLR) THEN !THE USER WANTS TO CLASSIFY R AS A BAD DATA VALUE !BRJ101115
MADE POLICY DECISION TO KEEP R AS OK, USER NEEDS TO WORRY ABOUT
VALIDA=.FALSE.
ELSE
VALIDA=.TRUE. !IGNORE R !BRJ101115 NOW HARD WIRED TO IGNORE, INCLR IS SET TO
FALSE
ENDIF
ENDIF
RETURN
END

SUBROUTINE ERRCHKA(STATN, YEAR, MONTH, JULDAY, HOUR,
1 OSTATN, OYEAR, OMONTH, OJULD, OHOUR, RECORD)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C BRJ101117 INSERTED STOP STATEMENT IF HOUR AND OHOUR CHECK FAILS
C BECAUSE CALLING PROGRAM SET TO GRAB 24H CHUNKS, IF HOUR IS MISSING
C THINGS GET MESSED UP AND MAY NOT BE OBVIOUS, SO JUST STOP THE PROGRAM
C AND MAKE THE USER FIX THE HOURLY PROGRESSION PROBLEM
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ERRCHKA - ERROR CHECKING. THIS ROUTINE CHECK TO MAKE SURE THAT
C THE YEAR REMAINS CONSTANT, AND THAT THE MONTHS, DAYS, AND HOURS
C ADVANCE IN NUMERICAL ORDER.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
INTEGER STATN, YEAR, MONTH, JULDAY, HOUR
INTEGER OSTATN, OYEAR, OMONTH, OJULD, OHOUR, RECORD
C
C SETUP OLD VALUES THE FIRST TIME THROUGH
IF (OSTATN .EQ. 0) OSTATN = STATN
IF (OYEAR .EQ. 0) OYEAR = YEAR
IF (OMONTH .EQ. 0) OMONTH = MONTH
IF (OJULD .EQ. 0) OJULD = JULDAY

IF (STATN .NE. OSTATN) THEN
WRITE (*,*) 'ERROR: STATION NUMBER CHANGED WITHIN INPUT FILE.'
WRITE (*,*) 'OLD: ', OSTATN, ' NEW: ', STATN
WRITE (*,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
WRITE (2,*) 'ERROR: STATION NUMBER CHANGED `WITHIN INPUT FILE.'
WRITE (2,*) 'OLD: ', OSTATN, ' NEW: ', STATN

```

```

WRITE (2,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
END IF

IF (YEAR .NE. OYEAR) THEN
WRITE (*,*) 'WARNING: YEAR CHANGED WITHIN INPUT FILE.'
WRITE (*,*) 'OLD: ', OYEAR, ' NEW: ', YEAR
WRITE (*,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
WRITE (2,*) 'WARNING: YEAR CHANGED WITHIN INPUT FILE.'
WRITE (2,*) 'OLD: ', OYEAR, ' NEW: ', YEAR
WRITE (2,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
C
C AND UPDATE OYEAR ON ASSUMPTION THAT THIS IS LEGITIMITE YEAR CHANGE
OYEAR=YEAR
CONTINUE
END IF
C
C MONTH MUST BE EQUAL OR GREATER BY 1
C EXCEPT WHEN YEAR CHANGES, JUST PRIN TOUT WARNING AND KEEP GOING

IF (MONTH .NE. OMONTH) THEN
IF (MONTH .NE. MOD(OMONTH,12)+1) THEN
WRITE (*,*) 'WARNING: MONTH OUT OF SEQUENTIAL ORDER.'
WRITE (*,*) 'OLD: ', OMONTH, ' NEW: ', MONTH
WRITE (*,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
WRITE (2,*) 'WARNING: MONTH OUT OF SEQUENTIAL ORDER.'
WRITE (2,*) 'OLD: ', OMONTH, ' NEW: ', MONTH
WRITE (2,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
OMONTH=MONTH
CONTINUE
ELSE
OMONTH = MONTH !UPDATE MONTH
END IF
END IF
C
C JULIAN DAY MUST BE EQUAL OR GREATER BY 1
C EXCEPT WHEN YEAR CHANGES
IF (JULDAY .NE. OJULD) THEN
IF ((JULDAY .NE. MOD(OJULD,365)+1).AND.
1 (JULDAY .NE. MOD(OJULD,366)+1)) THEN
WRITE (*,*) 'ERROR: JULIAN DAY OUT OF SEQUENTIAL ORDER'
WRITE (*,*) 'OLD: ', OJULD, ' NEW: ', JULDAY
WRITE (*,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
WRITE (2,*) 'ERROR: JULIAN DAY OUT OF SEQUENTIAL ORDER'
WRITE (2,*) 'OLD: ', OJULD, ' NEW: ', JULDAY
WRITE (2,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD

ELSE
OJULD = JULDAY
END IF
END IF
C
C HOUR MUST BE IN ORDER
IF (HOUR .NE. OHOUR+1) THEN
WRITE (*,*) 'ERROR: HOUR OUT OF SEQUENTIAL ORDER'
WRITE (*,*) 'OLD: ', OHOUR, ' NEW: ', HOUR
WRITE (*,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
WRITE (2,*) 'ERROR: HOUR OUT OF SEQUENTIAL ORDER'
WRITE (2,*) 'OLD: ', OHOUR, ' NEW: ', HOUR
WRITE (2,*) 'ERROR WAS ON OR BEFORE RECORD ', RECORD
WRITE(2,*) 'PROGRAM STOPPING BECAUSE HOUR SEQUENCE BAD' !BRJ101117
WRITE(2,*) 'USE RECORD NUMBER INFORMATION TO LOCATE' !BRJ101117
WRITE(2,*) 'AND FIX. HOURS MUST BE CONSECUTIVE ' !BRJ101117
WRITE(2,*) 'PROCESSING INCOMPLETE!' !BRJ101117
WRITE(*,*) 'PROGRAM STOPPING BECAUSE HOUR SEQUENCE BAD' !BRJ101117
WRITE(*,*) 'USE RECORD NUMBER INFORMATION TO LOCATE' !BRJ101117
WRITE(*,*) 'AND FIX. HOURS MUST BE CONSECUTIVE ' !BRJ101117
WRITE(*,*) 'PROCESSING INCOMPLETE!' !BRJ101117
STOP !BRJ101117

```

```

ELSE
  OHOUR = MOD(HOUR, 24)    !LOOP FROM 24 -> 0
END IF

END

      LOGICAL FUNCTION OPF1(ASC1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C OPENS FILE CONTAINING MET DATA AND CREATES OUTPUT FILE
C BASED ON INPUT FILE NAME
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      CHARACTER*11 ASC1
      CHARACTER*31 FILENM,F2,XNAME,F1
      CHARACTER*1 DOT
      LOGICAL FINDST
      INTEGER POS
      DOT='.'

      FILENM(12:31)=' '
      FILENM(1:11)=ASC1(1:11)
      WRITE(6,7979)FILENM(1:11),ASC1(1:11)
7979  FORMAT(1X,'LINE 617 FILENM= ',A11,' ASC1= ',A11)
      OPEN (UNIT=7, FILE=FILENM, STATUS='OLD', IOSTAT=IPERR)
      IF(IPERR.NE.0)GOTO23391
C
C USE NAME OF INPUT FILE AND ADD EXTENSION 'SUM'
      IF (.NOT. FINDST(DOT,1,FILENM,31,POS))THEN
      WRITE(6,1823)
1823  FORMAT(1X,'MUST ENTER FILENAME WITH PERIOD ')
      STOP
      ENDIF

      WRITE(6,1884)
1884  FORMAT(1X,'JUST AFTER FINDST ')
      XNAME(1:POS-1)=FILENM(1:POS-1)
      F1(1:POS-1)=XNAME(1:POS-1)
      F1(POS:POS+3)='.SUM'
      WRITE(6,7212)F1(1:11),FILENM(1:11),ASC1(1:11)
7212  FORMAT(1X,'F1= ',A11,' FILENM= ',A11,' ASCI= ',A11)

      OPEN(UNIT=2,FILE=F1,STATUS='NEW',IOSTAT=IPERR)
      IF(IPERR.NE.0)GOTO23391
      WRITE(2,1772)
1772  FORMAT(1X,'CIMPRO1V2 ....')
      WRITE(2,1773)FILENM
1773  FORMAT(1X,'INPUT DATA FILE IS ',A31,'".....')

      F2(1:POS-1)=XNAME(1:POS-1)
      F2(POS:POS+3)='.DET'
      OPEN(UNIT=8,FILE=F2,STATUS='NEW',IOSTAT=IPERR)
      IF(IPERR.NE.0)GOTO23391
      WRITE(2,8822)F2
8822  FORMAT(1X,'PROCESSED INPUT DATA FILE IS ',A31,'"...')
      OPF1=.TRUE.
      RETURN

23391  CONTINUE
      OPF1=.FALSE.
      WRITE(6,23377)IPERR,F1(1:11)
23377  FORMAT(1X,'UNABLE TO OPEN FILE ERR= ',I8,' , FILE= ',A11)
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      LOGICAL FUNCTION FINDST(SUB,SUBLEN,LINE,LINLEN,POS)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C FINDST RETURNS TRUE IF FINDS CHARACTER STRING SUB OF
C LENGTH SUBLIN IN CHARACTER STRING LINE OF LENGTH
C LINLEN. FIRST OCCURANCE OF SUB IN LINE OCCURS AT
C POS (POS RETURNED BY FINDST)
C
C **IMPORTANT MODIFICATION**FINDST MODIFIED TO IGNORE CASE
C OF LETTERS. ALL LETTERS CONVERTED TO UPPER CASE AND
C THEN STRING IS SEARCHED. 1/5/87
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER (A-Z)
      CHARACTER SUB
      CHARACTER*200 TSUB,TLINE
      CHARACTER LINE
C
C FALSE UNTIL PROVEN OTHERWISE
      FINDST=.FALSE.
C
C MODIFICATION 1/5/87 CONVERT ALL LOWER CASE TO UPPER CASE
C THEN DO SEARCH
      CALL FLUSHC(TSUB,100)
      CALL FLUSHC(TLINE,100)
      CALL CONVUL(SUB,TSUB,SUBLIN,'U')
      CALL CONVUL(LINE,TLINE,LINLEN,'U')
C
C MAKE SURE THAT SUBLIN IS LESS THAN LINLEN
      IF (SUBLIN.GT.LINLEN) THEN
100      WRITE (0,100)SUBLIN,LINLEN
          FORMAT(1X,'WARNING FINDST: S'
1          , 'UBLEN GT LINLEN'
2          ,I5,I5)
          RETURN
      ENDIF
C
C LOOK FOR SUBSTRING SUB IN LINE
      DO 10 I=1,LINLEN-SUBLIN+1
          IF (TSUB(1:SUBLIN).EQ.TLINE(I:I+SUBLIN-1)) THEN
              POS=I
              FINDST=.TRUE.
              RETURN
          ENDIF
10      CONTINUE
      RETURN
      END
      SUBROUTINE LOOK4E(LINE,COUNT,ERRUNI)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CIMIS EVIDENTLY HAS CHANGED HOW THEY REPORT MISSING VALUES, NOW THEY SEEM
C TO HAVE '--' AND THE REST OF THE FIELD IS MISSING THE BLANKS TO COMPLETE ITS SIZE
C SO, FOR EXAMPLE,      '*',      47.2,
C                        'M',--, ETC.
C SO I MANUALLY CHANGED THESE FIELDS TO BE PROPERLY EXPANDED FOR THE FILE TO BE
CONSISTENT
C ALSO, FOR PARLIER, I INTERPOLATED THE MISSING VALUES AND CHANGED 'M' TO 'E'.
C
C (UNFORTUNATELY, CIMIS ALSO USES E, BUT I'LL LIVE WITH THE AMBIGUITY, I LIKE E FOR
ESTIMATE)
C
C '--' IS RIGHT JUSTIFIED, SO WILL (1) ASSUME FIELD HAS BEEN EXPANDED (I.E. WAS ABLE TO
C RUN PREFORM - THIS IMPLIES SINCE PREFORM USES ALL FORTRAN FORMAT F FOR THE DATA READS
C THAT THERE ARE NO DOUBLE DASHES IN ANY DATA FIELDS, NEVETHELESSI WILL SET IT UP
C AS THOUGH THERE MIGHT BE AND IF SO, ALL PROCESSING HALTS AND THE LINE IS PRINTED OUT
C (THIS WILL BE OK FOR NOW, SINCE IT WON'T HAPPEN)
C
C SO AT LEAST FOR PARLIER, THIS PROGRAM WONT FIND ANY MISSING VALUES, HOWEVER, I LOOKED
C FOR ALL THE CODES USED IN PARLIER DATA AND FOUND 1 'S' IN THE TEMPERATURE DATA
CCCCCCCCCCCCCCCCCCCC END OF BRJ 070926 COMMENTS CCCCCCCCCCCCCCCCCC
C
C BEGINNING OF COMMENTS FROM OLDEN DAYS CCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C LOOK FOR BOTH '*****' AND '      --'
C
C '      --' 7 SPACES AND --.
C WHICH IS WHAT IS IN NEW CIMIS SYSTEM
C
C "LINE" IS 105 BYTE STRING WHICH HAS WEATHER LINE AND
C POTENTIALLY HAS 9 BYTE STRING OF ASTERISKS TO BE
C REPLACED BY -9999.000
C "COUNT" IS RECORD NUMBER OF LINE
C STRING LINE HAS '*****' REPLACED BY '-9999.000'
C THIS IS TO AVOID READ ERRORS
C "ERRUNI" IS ERROR UNIT OUTPUT
C IN ADDITION, LOOK4D CHECKS TO MAKE SURE THAT THE QUALITY CONTROL
C FLAG WHICH PRECEEDS '*****' IS NOT ' ' OR '*' OR 'F', WHICH
C WOULD INDICATE THAT THE VALUE WAS ACCEPTABLE. THIS SITUATION SHOULD
C NEVER HAPPEN, BUT CHECK FOR IT ANYWAY AND REPORT IT TO ERRUNI FILE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT INTEGER(A-Z)
      CHARACTER*105 LINE
      LOGICAL FINDST
C      CHARACTER*9 AST(2),CR
      CHARACTER*2 AST(2) !BRJ070926
      CHARACTER*4 CR      !BRJ070926
      CHARACTER DC

      AST(1)(1:2)='**'
      AST(2)(1:2)='--'
      CR(1:4)='-99.'

C      OPEN(UNIT=2,FILE='LOOK4C.OUT',STATUS='NEW')

      DO 955 IXK=1,2
C      IF(FINDST(AST(IXK),2,LINE,105,POS))THEN BRJ101021 REPLACE THIS WITH THE MODERN
INTRINSIC INDEX
                                                    !WHICH GIVES 0 IF DOESN'T FIND IT, AND
POSITION IF DOES
                                                    !WE WANT LEFT HAND POSITION HERE, USE
FALSE
      POS=INDEX(LINE,AST(IXK),.FALSE.)
      IF(POS.GT.0)THEN
1911      CONTINUE
          WRITE(6,6100)LINE
          WRITE(2,6100)LINE
          WRITE(ERRUNI,6100)LINE
6100      FORMAT(1X,'ERROR FROM LOOK4E, FOUND ** OR -- ',/1X,A104)
          STOP !BRJ101021 PROGRAM STOPS IF FINDS ** OR -- AND SUM FILE GETS LINE
PRINTOUT, MANUAL EDITING
          !IS SUPPOSED TO REMOVE, I COMMENTED OUT MOST OF THE LINES BELOW BECAUSE
I THINK THEY'RE
          !IRRELEVANT BRJ101021, I WAS GETTING AN ERROR USING THE FINDST
SUBROUTINE AND
          !SHOULD SWITCH OVER TO INDEX FUNCTION NOW.
      ENDIF
      !BRJ070926 AND THE PROGRAM IS STOPPED IF SOMETHING FOUND
C      !BRJ070926 FOLLOWING CODE MODIFIED FOR USE WITH NEW VARIABLE SIZES OF AST AND
CR ABOVE
C      !BRJ070926 EVEN THOUGH I'VE SET IT UP TO FAIL IF THESE THINGS ARE FOUND
C CHANGE '**' OR '--' TO '-99.'
C FIRST VERIFY THAT PRECEDING 2 CHARS ARE BLANK
C      IF(LINE(POS-2:POS-1).NE.' ')THEN
C      WRITE(ERRUNI,6200)LINE
C6200      FORMAT(1X,'LOOK4E: PRECEDING CHARS NOT BLANK ',/1X,A104)
C      STOP
C      ENDIF

C      LINE(POS-2:POS+1)=CR(1:4)

C MAKE SURE QUALITY CONTROL FLAG IS CONSISTENT

```

```

C      DC=LINE(POS-9:POS-9)
C      IF(DC.EQ.' '.OR.DC.EQ.'F'.OR.DC.EQ.'*')THEN
C      IF(.NOT.(DC.EQ.'M'.OR.DC.EQ.'I'.OR.DC.EQ.'S'.OR.DC.EQ.'R'))
C      *      THEN
C      WRITE(ERRUNI,6300)LINE
C6300      FORMAT(1X,'LOOK4E: FLAG - ERROR MISMATCH ',/1X,A104)
C      STOP !BRJ070926 YES STOP THE PROGRAM FOR NOW

```

C THIS SHOULD NEVER HAPPEN, BUT CHANGE DC TO 'R' AND REPORT IT

```

C      WRITE(ERRUNI,311)COUNT,LINE(1:104)
C      LINE(POS-9:POS-9)='R'
C      WRITE(ERRUNI,311)COUNT,LINE(1:104)
C311      FORMAT('BAD QUALITY CODE ON LINE ',I6,' WILL CHNG TO "R"',
C      1/1X,A104)
C      ENDIF

```

```

C      STCON=STCON+1 !BRJ070926 I HAVE NO IDEA WHAT THIS DOES
C      IF(FINDST(AST(IXK),9,LINE,105,POS))GOTO1911
C      ENDIF
955      CONTINUE
          RETURN
          END

```

```

          SUBROUTINE FLUSHC(STRING,LEN)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE FLUSHC SETS STRING OF LENGTH LEN ALL
C EQUAL TO BLANKS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          IMPLICIT INTEGER (A-Z)
          CHARACTER*200 STRING
C
          IF (LEN.EQ.0) THEN
              RETURN
          ELSE IF (LEN.LT.0) THEN
              WRITE(0,200)LEN
200          FORMAT(1X,'WARNING FROM FLUSHC: LEN < 0: LEN=',I4)
              RETURN
          ELSE
              IF (LEN.GT.120) THEN

                  ENDIF
                  DO 10 I=1,MIN(LEN,126)
10          STRING(I:I)=' '
                  RETURN
              ENDIF
          END

```

```

          SUBROUTINE CONVUL(STRNGI,STRNGO,LEN,DIRECT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C SUBROUTINE CONVUL CONVERTS STRINGI TO EITHER ALL UPPER CASE
C OR ALL LOWER CASE AND PUTS INTO STRNGO. DIRECT IS SINGLE
C BYTE CHARACTER 'U'=CONVERT TO UPPER 'L'= CONVERT TO LOWER
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          IMPLICIT INTEGER(A-Z)
          CHARACTER*200 STRNGO,STRNGI
          CHARACTER*1 DIRECT
C
          IF (DIRECT.EQ.'U') THEN
              DO 100 I=1,LEN
                  IF (ICHAR(STRNGI(I:I)).GE.97.AND.ICHAR(STRNGI(I:I)).LE.122)
1          THEN
                      STRNGO(I:I)=CHAR(ICHAR(STRNGI(I:I))-32)
                  ELSE
                      STRNGO(I:I)=STRNGI(I:I)

```



```

        ENDIF
100    CONTINUE
C
    ELSE IF (DIRECT.EQ.'L') THEN
        DO 200 I=1,LEN
            IF (ICHAR(STRNGI(I:I)).GE.65.AND.ICHAR(STRNGI(I:I)).LE.90)
2          THEN
                STRNGO(I:I)=CHAR(ICHAR(STRNGI(I:I))+32)
            ELSE
                STRNGO(I:I)=STRNGI(I:I)
            ENDIF
200    CONTINUE
C
    ELSE
        WRITE(0,300)DIRECT
300    FORMAT(1X,'CONVUL: WARNING WRONG VALUE FOR DIRECT = ',A1)
        ENDIF
C
        RETURN
        END
        LOGICAL FUNCTION ASK(MESS,LEN)
C
C  TAKES MESSAGE IN ARRAY MESS OF LENGTH N AND ASKS YES NO QUESTION
C  OF FORM --ASK QUESTION?(Y/N)
C  THEN IT READS IN Y/N ANSWER AND SETS ITSELF TRUE OR FALSE
C
        IMPLICIT INTEGER (A-Z)
C        BYTE MESS(50),FORM(30),ANS
C        ENCODE (30,100,FORM)LEN
C100    FORMAT ('('$--',I2,'A1','?(Y/N) '\)')
CD      WRITE (5,200)LEN,FORM
CD200   FORMAT (1X,'LEN=',I2,'FORM= ',30A1)
C
C  PREVIOUS COMMENTED LINES LEFT IN FOR HISTORICAL PURPOSES
C  AS THEY WAY IT WAS WITH ENCODE STATEMENTS ON THE PDP
        CHARACTER*50 MESS
        CHARACTER*30 FORM
        CHARACTER*1 ANS
C
C  CREATE APPROPRIATE FORMAT
        WRITE (FORM,100)LEN
100    FORMAT('1X,'--',A',I2,'','?(Y/N) '\)')
c      write (0,150)form
c150   format(a30)
2      CONTINUE
        WRITE (6,FORM)MESS(1:LEN)
        READ (5,350)ANS
350    FORMAT (A1)
        IF (ANS.EQ.'Y'.OR.ANS.EQ.'y')ASK=.TRUE.
        IF (ANS.EQ.'N'.OR.ANS.EQ.'n')ASK=.FALSE.
        IF (ANS.EQ.'N'.OR.ANS.EQ.'Y'.OR.ANS.EQ.'n'.or.ANS.EQ.'y')RETURN
        WRITE (6,400)
400    FORMAT(1X,'TYPE A "Y" OR AN "N" TO ANSWER QUESTION ')
        GO TO 2
        END
        SUBROUTINE INTRO
        CHARACTER*1 a
        call men01()
        read(5,10) a
10     FORMAT(a1)
        call men02()
        READ(5,10) a
        call men03()
        READ(5,10)a
        return
        end
        SUBROUTINE men01
C
        WRITE(6,1900)
1900   FORMAT(//1X,'CIMPRO1V2 COMPILED 9/27/2007',
        */1X,' ',

```

```

*/IX,'CIMPRO1V2 is based on CIMPRO1.FOR, which does phas',
*'e l',
*/IX,'analysis of downloaded CIMIS data. Downloaded CIM',
*'IS data',
*/IX,'now has a different format than when CIMPRO1 was o',
*'riginally',
*/IX,'written. Consequently, it is necessary to run PRE',
*'FORM.FOR on')
WRITE(6,2000)
2000 FORMAT(1X,'current CIMIS downloads. PREFORM reformats CIMIS ',
*'downloads into',
*/IX,'the prior format (this was easier than rewriting C',
*'IMPRO1 to',
*/IX,'accomodate the new format).',
*/IX,' ',
*/IX,'The downloaded variables are: solar radiation (ly/',
*'h), net radiation (ly/h)',
*/IX,'air temperature (F), wind velocity (MPH), wind dir',
*'ection (deg), standard')
WRITE(6,2100)
2100 FORMAT(1X,'deviation of wind direction (deg), precipitation (',
*'inches).',
*/IX,' ',
*/IX,'--Press <Enter> to continue--',
*/IX,' ')
RETURN
END
SUBROUTINE men02
C
WRITE(6,1900)
1900 FORMAT(1X,'CIMPRO1V2 expects the following file format:',
*/IX,' ',
*/IX,'      1      2      3      4      5',
*'      6',
*/IX,'12345678901234567890123456789012345678901234567890',
*'1234567890123456789',
*/IX,'stid yy mm      hh jd      q netradnet q air',
*'tempaa q windveloc',
*/IX,'0039,06-01-01,01,001,*,      0.000,*,      -35.000,*,      ',
*'42.900,*,      1.400,')
WRITE(6,2000)
2000 FORMAT(1X,' ',
*/IX,' ',
*/IX,'7      8      9      0',
*/IX,'012345678901234567890123456789012345678901234',
*/IX,'q winddirwi q windsdwin',
*/IX,'*,      36.400,*,      22.900,*,      0.000',
*/IX,' ',
*/IX,'stationid,year,month,hour,julianday,net radiation,',
*'air temp,wind speed,')
WRITE(6,2100)
2100 FORMAT(1X,'wind direction, standard deviation of wind directi',
*'on',
*/IX,' ',
*/IX,'q=quality control flag for following field',
*/IX,' ',
*/IX,'CIMPRO1V2 ignores solar radiation and precipitatio',
*'n, but formatting',
*/IX,'requires those fields be present.')
```

```

WRITE(6,2200)
2200 FORMAT(1X,' ',
*/IX,'--Press <Enter> to continue--',
*/IX,' ')
RETURN
END
SUBROUTINE men03
C
WRITE(6,1900)
1900 FORMAT(1X,'CIMPRO1V2 produces two files: "name".OUT and "name"',
*' .DET where',
*/IX,'"name" = the name of the input file (without exten',

```

```

*'sion).',
*/1X,' ',
*/1X,'name.out contains the original data file with "#" ',
*'in column 105',
*/1X,'      for the days which would be excluded unde',
*'r the')
WRITE(6,2000)
2000 FORMAT(1X,'      criteria (criteria in second file)',
*/1X,' ',
*/1X,'name.DET contains a summary of the bad data values',
*' for each day',
*/1X,'      where there was at least one bad value',
*/1X,' ',
*/1X,'Bad values are defined as those fields which have ',
*'a QC flag of',
*/1X,'M,S,I,R except for wind speed (probably ws=1.0, ',
*/1X,'and "R" is user-determined as to bad or not/')
WRITE(6,2100)
2100 FORMAT(1X,'M=missing, S=sensor not working or data out of sen',
*'sor range',
*/1X,'I=ignore value (all windspeeds of 1mph are flagged',
*' with I)',
*/1X,'R=Data far out of historical limits.',
*/1X,' ',
*/1X,'Press <Enter> to continue')
RETURN
END
SUBROUTINE IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CALLING PROGRAM NEEDS TO DECLARE THE FOLLOWING
C-----
C CHARACTER*35 FIN,FOUT
C LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
C INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
C UIN=1
C UOUT=2
C-----
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
INTEGER UIN,UOUT
CHARACTER*35 FIN,FOUT
LOGICAL INSTATE,OUTSTATE,INSTATUS
CHARACTER*70 CLINE

CALL GETCL(CLINE)
CALL IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)

IF(INSTATE.AND.OUTSTATE)THEN
C WRITE(6,500)FIN,FOUT,K1,K2
C500 FORMAT(1X,'IOREADY: INPUT,OUTPUT',/1X,A35,/1X,A35,
C * /1X,'K1,K2 ',2I3)
CONTINUE !STUFF ABOVE FOR DEBUGGING
ELSEIF (.NOT.INSTATE)THEN
WRITE(6,505)FIN,K1
505 FORMAT(1X,'IOREADY: INPUT FILENAME ERROR ',A35,1X,I3)
STOP
ELSEIF (.NOT.OUTSTATE)THEN
WRITE(6,510)FOUT,K2
510 FORMAT(1X,'IOREADY: OUTPUT FILENAME ERROR ',A35,1X,I3)
STOP
ENDIF
!CHECK TO MAKE SURE THE INPUT FILE EXISTS
INQUIRE(FILE=FIN,EXIST=INSTATUS)
IF(.NOT.INSTATUS)THEN
WRITE(6,255)FIN
255 FORMAT(1X,'IOREADY: INPUT FILE NOT EXIST',
* /1X,A35, /1X,'PROGRAM STOPPING ')
STOP
ENDIF
OPEN(UNIT=UIN,STATUS='OLD',FILE=FIN,ERR=1000)

```

```

OPEN(UNIT=UOUT,STATUS='UNKNOWN',FILE=FOUT,ERR=2000)
RETURN

!ERRORS ON OPENING GO BELOW
1000 CONTINUE
WRITE(6,1100)FIN
1100 FORMAT(1X,'ERROR TRYING TO OPEN IN FILE ',/1X,A35)
STOP
2000 CONTINUE
WRITE(6,2100)FOUT
2100 FORMAT(1X,'ERROR TRYING TO OPEN OUT FILE ',/1X,A35)
STOP
END SUBROUTINE

SUBROUTINE IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)
IMPLICIT NONE
CHARACTER*35 FIN,FOUT
CHARACTER*70 CLINE
INTEGER K1,K2,IBLANK,K,I,L
LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
!K1, K2 ARE LENGTHS OF THE FILENAMES

DO I=1,35
  FIN(I:I)=' '
  FOUT(I:I)=' '
END DO
!ASSUME CLINE HAS 1 OR MORE BLANK SPACES WHICH DIVIDE THE TWO FILENAMES
!ASSUME THAT INPUT FILE IS THE FIRST FILE, AND OUTPUTFILE IS THE SECOND ONE
K=LEN_TRIM(CLINE) !FIND OUT TOTAL STRING LENGTH
!LOOK FOR FIRST BLANK
IBLANK=0
DO I=1,K
  IF(CLINE(I:I).EQ.' ')THEN
    IBLANK=I
    GOTO100
  ELSE
    CONTINUE
  ENDIF
END DO
100 IF (IBLANK.EQ.0) THEN
  INSTATE=.FALSE.
  OUTSTATE=.FALSE.
  K1=0
  K2=0
  WRITE(6,200)CLINE
200 FORMAT(1X,'COMMAND LINE LACKING 2 FILENAMES ',/1X,A70)
  RETURN
ELSEIF (IBLANK.GE.70)THEN
  WRITE(6,250)CLINE
250 FORMAT(1X,'FILENAME TOO LONG ',/1X,A70)
  K1=IBLANK
  K2=0
  INSTATE=.TRUE.
  OUTSTATE=.FALSE.
ELSE
  FIN(1:IBLANK-1)=CLINE(1:IBLANK-1)
  L=K-(IBLANK-1)
  FOUT(1:L)=CLINE(IBLANK:K)
  FOUT(1:L)=ADJUSTL(FOUT(1:L))
  K2=LEN_TRIM(FOUT(1:L))
  IF(K2.EQ.0)THEN
    INSTATE=.TRUE.
    OUTSTATE=.FALSE.
    RETURN
  ENDIF
  K1=IBLANK-1
  INSTATE=.TRUE.
  OUTSTATE=.TRUE.
ENDIF
RETURN
END SUBROUTINE

```

```

      SUBROUTINE IOSUPP(OUT2STATE,U2OUT,FOUT,F2OUT)
      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      C
      C THIS SUBROUTINE REPLACES A PORTION OF OPF1 SUBROUTINE IN
      C EARLIER VERSION OF CIMPRO1V2. OPF1 OPENED THE INPUT AND OUTPUT FILES
      C BASED ON READING A FILENAME FROM UNIT 13 WHICH AS ASCLIST.DAT
      C
      C NOW I'M CHANGING THE INPUT/OUTPUT MECHANICS TO CIMPRO1 INFILE OUTFILE
      C BUT TRYING TO MAKE THESE CHANGES WITHOUT CHANGING ANY OF THE GUTS OF THE
      C PROGRAM, JUST SORT OF MAKE THE CHANGES AND LEAVE THE REST OF THE PROGRAM ALONE
      C WITH THE SAME INPUT AND OUTPUT UNIT DESIGNATORS (WILL NO LONGER NEED UNIT 13)
      C
      C ANOTHER THING THAT OPF1 DID WAS TO WRITE THE CIMPRO1 VERSION NUMBER INTO THE
      C 'SUM' FILE AND I WILL HAVE THAT FEATURE INCLUDED JUST BELOW WHERE IOSUPP IS CALLED IN
      C THE M
      C THE MAIN PROGRAM NSTEAD OF BURIED.
      C
      C IOSUPP REQUIRES THE FOLLOWING DECLARATIONS IN THE CALLING ROUTINE
      C
      C LOGICAL OUT2STATE !SUCCESS = TRUE, FAILURE = FALSE
      C INTEGER U2OUT !OUTPUT UNIT, WHICH IN CALLING PROGRAM WILL PROBABBLY BE 2
      C CHARACTER*35 FOUT !OUTPUT FILENAME ALREADY OPENED (BJ1 FILENAME)
      C CHARACTER*39 F2OUT !OTUPUT FILE CREATED IN IOSUPP, WITH .SUM EXTENSION
      C
      C IOSUPP RETURNS OUT2STATE AS TRUE IF FILE F2OUT SUCCESSFULLY OPENED AND FALSE OTHERWISE
      C WITH THE FILENAME IN F2OUT (UP TO 39 CHARACTERS
      IMPLICIT NONE
      INTEGER I,ILEN
      INTEGER U2OUT
      LOGICAL OUT2STATE
      CHARACTER*35 FOUT
      CHARACTER*39 F2OUT

      ILEN=LEN_TRIM(FOUT) !THIS SHOULD BE OK FROM IOREADY PROCESSING
      DO I=1,39
      F2OUT(I:I)=' '
      END DO
      IF (ILEN.EQ.0) THEN
      WRITE(6,100)
      100   FORMAT(1X,'IOSUPP: FOUT HAS ZERO LENGTH....')
      OUT2STATE=.FALSE.
      RETURN
      ENDIF
      !LOOK FOR A PERIOD, THE INSTRUCTIONS WILL TELL PEOPLE TO USE EXTENSION, BUT
      !YOU NEVER KNOW
      I=SCAN(FOUT(1:ILEN),'.',.FALSE.) !GET MODERN, USE INTRINSIC SCAN
      IF(I.EQ.0) THEN !NO PERIOD FOUND, JUST ADD '.SUM'
      F2OUT(1:ILEN)=FOUT(1:ILEN)
      F2OUT(ILEN+1:ILEN+4)='.SUM'
      ELSE
      F2OUT(1:I-1)=FOUT(1:I-1)
      F2OUT(I:I+3)='.SUM'
      ENDIF
      OPEN(UNIT=U2OUT,FILE=F2OUT,STATUS='UNKNOWN',ERR=1000)
      OUT2STATE=.TRUE.
      RETURN

      1000  CONTINUE !SOME KIND OF ERROR ON THE OPEN
      WRITE(6,1100)F2OUT
      1100  FORMAT(1X,'IOSUPP: ERROR OPENING F2OUT ',/1X,A39)
      OUT2STATE=.FALSE.
      RETURN

      END SUBROUTINE

```

CIMPRO2V5.FOR

```
C      Last change:  BJ   23 Jan 2014   1:06 pm
PROGRAM CIMPRO2v5
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C 121312 TERRI FOUND CASES WHERE NIGHTTIME STABILITY CLASSES OCCURRED DURING DAYTIME HORUS
C CIMIS DATA CONTAINS NEGATIVE NET RADIATION FOR DAYTIME HOURS OCCASIONALLY.  CIMIS
PERSONNEL HAVE NO
C EXPLANATION FOR THIS AND THE VALUES ARE NOT FLAGGED AS INCORRECT.  SO THESE PROGRAMS
WOULD GO AHEAD
C AND USING NIGHTTIME STABIOLITY ROUTINES .  I HAVE ADDED A FUNCTION WHICH CHECKS FOR
NEGATIVE NET
C RADIATION DURING THE DAY AND CHANGES IT TO A POSITIVE NUMBER (ALSO IF ZERO, CHANGES IT
TO SMALL
C POSITIVE VALUE), NET RAD IS ONLY USED IN THESE PROGRAMS TO DETERMINE NIGHT AND DAY.
C
C FUNCTION INSERTED IS READ FUNCTION FIXNETRAD(M,H,NR), WHERE M=MONTH,H=HOUR,NR=CIMIS NET
RADIATION
C IF THE RADIATION IS NEGATIVE DURING THE DAY (DAY DEFINED BY TABLE OF SUNRSIE SUNSET
HOURS FROM
C SAN DIEGO AND REDDING) THEN IT'S CHANGED TO POSITIVE.  THE 8S SHOWS WHERE FIXNETRAD
DEFINES DAY
C
C      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
C 1  1  1  1  1  1  1  1  1  1  1  1  1
C 2  1  1  1  1  1  1  1  1  1  1  1  1
C 3  1  1  1  1  1  1  1  1  1  1  1  1
C 4  1  1  1  1  1  1  1  1  1  1  1  1
C 5  1  1  1  1  1  1  1  1  1  1  1  1
C 6  1  1  1  1  1  1  1  1  1  1  1  1
C 7  1  1  1  8  8  8  8  8  8  1  1  1
C 8  1  1  8  8  8  8  8  8  8  8  8  1
C 9  1  8  8  8  8  8  8  8  8  8  8  8
C 10 8  8  8  8  8  8  8  8  8  8  8  8
C 11 8  8  8  8  8  8  8  8  8  8  8  8
C 12 8  8  8  8  8  8  8  8  8  8  8  8
C 13 8  8  8  8  8  8  8  8  8  8  8  8
C 14 8  8  8  8  8  8  8  8  8  8  8  8
C 15 8  8  8  8  8  8  8  8  8  8  8  8
C 16 1  8  8  8  8  8  8  8  8  8  1  1
C 17 1  1  1  1  8  8  8  8  1  1  1  1
C 18 1  1  1  1  1  1  1  1  1  1  1  1
C 19 1  1  1  1  1  1  1  1  1  1  1  1
C 20 1  1  1  1  1  1  1  1  1  1  1  1
C 21 1  1  1  1  1  1  1  1  1  1  1  1
C 22 1  1  1  1  1  1  1  1  1  1  1  1
C 23 1  1  1  1  1  1  1  1  1  1  1  1
C 24 1  1  1  1  1  1  1  1  1  1  1  1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C SO THE SCHEME REFLECTS SEASONS
C ALSO CHANGED THRESHOLD BACK TO 1.25 TO KEEP CONSISTENCY
C CHANGED BJ1 TO DET AND 2P0 TO 2PC, AVOID 0 OR O CONFUSION
C
C
C FABIO FOUND BUG WHERE WHEN THE INITIAL STABILITY CLASS WAS E,
C AND THE WIND SPEED WAS >5M/S, THE PROGRAM INCORECTLY ASSIGNED
C STABILITY CLASS OF F, SHOULD HAVE BEEN D (SEE N:\UPDATES\120806FIXSTABBUG UNDER PARLIER
CIMIS DIRECTORY)
C
C THIS VERSION WILL BE CIMPRO2V4 (TO REPLACE CIMPRO2V3).
C
C
C CIMPRO2V3 MODIFIED CIMPRO2V2A AS FOLLOWS:
C 1. REPLACED OPENFL WITH IOREADY SUBROUTINE
C A. NOTE THAT INPUT IS UNIT 8 AND OUTPUT IS UNIT 9
C 2. WILL CHANGE STATUS OF OUTPUT OPEN TO 'UNKNOWN' FROM 'NEW', SO THAT IT OVERWRITES
C 3. EDGAR AND I DECIDED AFTER EXAMINING THE R FLAGGED VALUES IN SEVERAL FILES THAT
```

```

C      WE WOULD INCLUDE R FLAGGED VALUES, THIS WILL NOW BE WRITTEN UP INTO THE
INSTRUCTIONS
C      AND THERE ARE POTENTIAL PROBLEMS WITH THE WIND DIRECTION BECAUSE WHEN THOSE GOT
FLAGGED
C      WITH R IT SEEMS LIKE THE DIRECTIONS WERE CONSTANT FOR SEVERAL HOURS.  BUT THIS
PROBLEM
C      WILL HAVE TO BE ADDRESSED BY THE USER
C      IN TERMS OF PROGRAMMING, THE FLAG 'INCLR' WILL BE SET TO FALSE IN ORDER TO GET R-
FLAGGED
C      VALUES INCLUDED IN THE MET DATA.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C BRJ100715 CHANGED THRESHOLD FROM 2.8 TO 2 MPH BECAUSE NATURE OF THESE SIMULATIONS
C IS LOCALIZED.
C
C BRJ071001 CHANGED THE REFERENCES TO INDEX WHICH WERE IN HERE ORIGINALLY TO INDEXP
C IN ORDER NOT TO CONFLICT WITH FORTRAN INTRINSIC FUNCTION INDEX (NEW THING SINCE LAST
C WORKED ON THIS)
C
C BRJ070928 THIS IS MODIFIED FROM CIMPRO2, WHICH IS PHASE 2 OF CIMIS DATA
C PROCESSING AND (1) ESTIMATES MISSING VALUES (2) ESTIMATES STABILITY
C (3) WRITES OUT FILE WITH ISC COMPATIBLE FORMAT FOR MODELING
C
C PROGRAM ASKS AT BEGINNING WHETHER TO INCLUDE 'R' AS A BAD DATA ITEM.
C CIMPRO1V2 ALSO ASKS THAT QUESTION, HANDLING THIS ISSUE SHOULD BE CONSISTENT
C BETWEEN THE TWO PROGRAMS -- SEE DOCUMENTATION OF CIMPRO1V2 FOR
C DIFFERENCES IN THE 'NEW' CIMIS OUTPUT AND THE OLD.  NOTE: THE NEW PROGRAM
C PREFORM.FOR, SHOULD HAVE BEEN USED INITIALLY TO GET THE DOWNLOADED DATA INTO
C THE OLD FORMAT.  ALSO, DATA SHOULD BE DOWNLOADED IN ENGLISH UNITS
C SO FOR INITIAL PHASE (1) RUN PREFORM (2) RUN CIMPRO1V2, THEN YOU MAKE
C DECISIONS ABOUT WHAT TO DO WITH 'R' VALUES OR ANY OTHER MISSING VALUES
C AND RUN CIMPRO2V2
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C TAKES OUTPUT FROM CIMPRO1, AND PRODUCES ISCST3 FORMATTED MET FILE FOR
C USE IN MODELING
C234567890123456789012345678901234567890123456789012345678901234567890123456789
01234567890
C016,92-01-01,01, 1,*, -3.306,*, -69.174,*, 45.860,*, 8.083,*, 134.300,*,
5.778,*, 0.000
C          Q SOL RAD Q NET RAD Q TEMP Q WVEL Q WDIR Q SD
Q PRECIP
C CIMIS
C SENSOR
C NUMB          Q R (F9.3)
C 1 SOLAR RAD ENGLISH LY/DAY [W/M^2] 22 24
C 2 NETRAD NET RADIATION ENGLISH LY/DAY [W/M^2] 34 36
C 4 AIRT AIR TEMPERATURE ENGLISH FAHRENHEIT [DEG C] 46 48
C 8 WVEL WIND VELOCITY ENGLISH MILES PER HOUR [M/S] 58 60
C 10 WDIR WIND DIRECTION [0-360 DEG] 70 72
C 11 SIGMAT SIGMA-THETA STD DEV OF WIND DIRECTION [DEG] 82 84
C 12 PRECIPITATION INCHES 94 96
C
C
INTEGER STATN, YEAR, MONTH, JULDAY, HOUR
REAL NETRAD, AIRT, WVEL, WDIR, SIGMAT

CHARACTER NETRAQ, AIRTQ, WVELQ, WDIRQ, SIGMAQ
CHARACTER USABLE(24)
CHARACTER*8 NAME
CHARACTER*12 FN1
CHARACTER*105 CF(24) ! ORIGINAL LINE
CHARACTER*56 BLANK !USE FOR DAYS WITH UNUSABLE DATA TO FILL OUT LINE

CHARACTER*80 COMLIN !COMMAND LINE FILENAME ARG
COMMON /COMAND/COMLIN !COMMUNICATE WITH SUBROUTINE
LOGICAL UZAB !IF # IN 105 TAKE APPROPRIATE ACTION

INTEGER OSTATN, OYEAR, OMONTH, OJULD, OHOURL, OSTAB

```

```

REAL      OWDIR

INTEGER   STABCL, RECORD
REAL     WDIR10, MIXHT
LOGICAL   HEADER

DIMENSION STATN(24), YEAR(24), MONTH(24), JULDAY(24), HOUR(24)
DIMENSION NETRAD(24), AIRT(24), WVEL(24), WDIR(24), SIGMAT(24)
DIMENSION NETRAQ(24), AIRTQ(24), WVELQ(24), WDIRQ(24), SIGMAQ(24)
DIMENSION STABCL(24), WDIR10(24), MIXHT(2,24), OWDIR(24)

LOGICAL   DEBUG1, DEBUG2, DEBUG3, DEBUG4

COMMON /REJCRI/RC,INCLR      !BRJ070927
INTEGER RC                   !REJECTION CRITERIA NOT USED IN THIS PROGRAM, BUT KEEP FOR
EASY OF PROGRAMMING

LOGICAL ASK,INCLR           !BRJ070927
CHARACTER*10 ERRMESS !BRJ070927 ADDED TO TIP OFF VALIDA WHEN WIND VELOCITY IS BEING
EVALUATED

CHARACTER*35 FIN,FOUT      !BRJ101025 IOREADY ARGUMENTS
LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS

REAL FIXNETRAD,FIXNR      !BRJ131213  INSERTED REAL FUNCTION FIXNETRAD TO MAKE SURE NO
NEGATIVE DAYTIME NET RADIATION

PARAMETER (DEBUG1 = .FALSE.)
PARAMETER (DEBUG2 = .FALSE. )
PARAMETER (DEBUG3 = .FALSE. )
PARAMETER (DEBUG4 = .FALSE. )

RC=0 !THIS IS NOT USED BRJ070927
UIN=8 !BRJ101025 IOREADY ARGUMENTS
UOUT=9 !BRJ101025 IOREADY ARGUMENTS

C      CALL GETCL(COMLIN) !GET MET DATA FILENAME      !BRJ101025 COMMENTED THIS OUT, NO
LONGER NEEDED WITH IOREADY
      BLANK(1:28)='
      BLANK(29:56)=BLANK(1:28)

C      CALL OPENFL !UNIT8= INPUT MET FILE FROM ABOVE, UNIT9=OUTPUT !BRJ101025 COMMENTED
OUT, REPLACED BY IOREADY ROUTINE
C      IF(ASK('DO YOU WANT "R" FLAG AS BAD VALUE ',34))THEN      !BRJ070927
C      INCLR=.TRUE. !TRUE MEANS R WILL BE INCLUDED WITH I,M,S AS BAD VALUES
C      ELSE
      INCLR=.FALSE. !BRJ101025 REST OF IF,THEN COMMENTED OUT, POLICY WILL BE INCLUDE
R
C      ENDIF
      CALL IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT) !BRJ101025 INSERT NEW
IOSUBROUTINE HERE
      IF(.NOT.(INSTATE.AND.OUTSTATE))THEN
      WRITE(6,1993)
1993  FORMAT(1X,'CIMPRO2: PROBLEM WITH FILES, PROGRAM STOPPING ')
      STOP
      ENDIF

C      INITIALIZE A FEW THINGS
RECORD = 0
OSTATN = 0
OYEAR = 0
OMONTH = 0
OJULD = 0
OSTAB = 0

DO I=1,24
OWDIR(I) = 0.0
END DO

```



```

DO WHILE (.TRUE.)

DO I=1,24
  READ(8,8903,END=997,ERR=998)CF(I)
8903  FORMAT(A105)
      READ (CF(I),9101, END=997, ERR=998) STATN(I), YEAR(I), MONTH(I),
1HOURL(I), JULDAY(I), NETRAQ(I), NETRAD(I), AIRTQ(I), AIRT(I),
2WVELQ(I), WVEL(I),WDIRQ(I), WDIR(I), SIGMAQ(I), SIGMAT(I),
3USABLE(I)
9101  FORMAT(I4, T6, I2, T9, I2, T15, I2, T18, I3, T34, A1, T36, F9.0,
1T46, A1, T48, F9.0, T58, A1, T60, F9.0, T70, A1, T72, F9.0,
2T82,A1, T84,F9.0,T105,A1)

RECORD = RECORD + 1
END DO

C      STEP 2.0 - CHECK FOR NONUSABILITY CHARACTER IN COLUMN 105,
C      CHARACTER IS '#'. IF UNUSABLE, THEN
C      SKIP THE REST OF THE CODE AND GO TO THE END OF READ LOOP

      UZAB=.TRUE.
      IF(USABLE(1).EQ. '#')THEN
        UZAB=.FALSE.
        GOTO1891
      ENDIF

C      STEP 2.1 - CHECK FOR BAD DATA OR CALM WINDS

C      THIS ROUTINE FILLS IN BAD VALUES WITH NEIGHBORING VALUES

      IF (DEBUG1) THEN
        WRITE (*,*) '_____ '
        WRITE (*,*) 'NET RADIATION BEFORE'
        WRITE (*,*)
        DO I=1,24
          WRITE (*,*) NETRAD(I), ' ', NETRAQ(I), ' ', RECORD
        END DO
        END IF

C      1234567890
      ERRMESS='XXXXXXXXXX' !BRJ070927
      CALL CHECKQ(NETRAD, NETRAQ, RECORD, DEBUG2,ERRMESS) !BRJ070927 MODIFIED ADDED
ERRMESS ARGUMENT

      IF (DEBUG1) THEN
        WRITE (*,*) '_____ '
        WRITE (*,*) 'NET RADIATION AFTER'
        WRITE (*,*)
        DO I=1,24
          WRITE (*,*) NETRAD(I), ' ', NETRAQ(I), ' ', RECORD
        END DO
        WRITE (*,*) '_____ '
        WRITE (*,*) 'AIR TEMPERATURE BEFORE'
        WRITE (*,*)
        DO I=1,24
          WRITE (*,*) AIRT(I), ' ', AIRTQ(I), ' ', RECORD
        END DO
        END IF

      CALL CHECKQ(AIRT, AIRTQ, RECORD, DEBUG2,ERRMESS) !BRJ070927

      IF (DEBUG1) THEN
        WRITE (*,*) '_____ '
        WRITE (*,*) 'AIR TEMPERATURE AFTER'
        WRITE (*,*)
        DO I=1,24
          WRITE (*,*) AIRT(I), ' ', AIRTQ(I), ' ', RECORD
        END DO
        WRITE (*,*) '_____ '

```

```

WRITE (*,*) 'WIND VELOCITY BEFORE'
WRITE (*,*)
DO I=1,24
WRITE (*,*) WVEL(I), ' ', WVELQ(I), ' ', RECORD
END DO
END IF

```

```

ERRMESS='WINDVEL ' !BRJ070927
CALL CHECKQ(WVEL, WVELQ, RECORD, DEBUG2,ERRMESS) !BRJ070927

```

```

IF (DEBUG1) THEN
WRITE (*,*) '_____ '
WRITE (*,*) 'WIND VELOCITY AFTER'
WRITE (*,*)
DO I=1,24
WRITE (*,*) WVEL(I), ' ', WVELQ(I), ' ', RECORD
END DO
WRITE (*,*) '_____ '
WRITE (*,*) 'WIND DIRECTION BEFORE'
WRITE (*,*)
DO I=1,24
WRITE (*,*) WDIR(I), ' ', WDIRQ(I), ' ', RECORD
END DO
END IF

```

```

ERRMESS='XXXXXXXXXX'
CALL CHECKQ(WDIR, WDIRQ, RECORD, DEBUG2,ERRMESS) !BRJ070927

```

```

IF (DEBUG1) THEN
WRITE (*,*) '_____ '
WRITE (*,*) 'WIND DIRECTION AFTER'
WRITE (*,*)
DO I=1,24
WRITE (*,*) WDIR(I), ' ', WDIRQ(I), ' ', RECORD
END DO
WRITE (*,*) '_____ '
WRITE (*,*) 'SIGMA T BEFORE'
WRITE (*,*)
DO I=1,24
WRITE (*,*) SIGMAT(I), ' ', SIGMAQ(I), ' ', RECORD
END DO
END IF

```

```

CALL CHECKQ(SIGMAT, SIGMAQ, RECORD, DEBUG2,ERRMESS) !BRJ070927

```

```

IF (DEBUG1) THEN
WRITE (*,*) '_____ '
WRITE (*,*) 'SIGMA T AFTER'
WRITE (*,*)
DO I=1,24
WRITE (*,*) SIGMAT(I), ' ', SIGMAQ(I), ' ', RECORD
END DO
END IF

```

C
C
C

C STEP 3 - MAKE NECESSARY CONVERSIONS
C CONVERT FIRST FROM ENGLISH TO METRIC

```

DO I=1,24

```

C CONVERT AIR TEMPERATURE FROM CELCIUS TO KELVIN
C AFTER CONVERTING FROM FAHRENHEIT TO CELSIUS

```

AIRT(I)=(AIRT(I)-32.)*5./9.
AIRT(I) = AIRT(I) + 273.15

```

C
C CHECK FOR WIND SPEEDS BELOW 2.8 MILES PER HOUR BASED ON APPENDIX 2


```

        LWVEL(I), AIRT(I), WDIR(I), MIXHT(1,I), MIXHT(2,I)          !BRJ070927 eliminate
wdir10, not used anymore
        END DO
        END IF

201  FORMAT(I3, I3, F6.2, I5, 5F10.2)

1891  CONTINUE !THIS IS JUMP FROM WHEN '#' IS DETECTED IN COLUMN 105
      IF(UZAB)THEN !THIS IS A USABLE RECORD AND CALCULATIONS HAVE BEEN MADE
        DO 1515 L=1,24
          CALL FIXER(CF(L),NETRAD(L),AIRT(L),WVEL(L),WDIR(L),SIGMAT(L))

C-----
CNOW FORMAT THE RIGHT SIDE OF THESE RECORDS FOR ISCST3
C-----
CME INPUTFIL SBP1.MET (4I2,2F9.4,F6.1,I2,2F7.1)
CE ANEMHGHT 10.000 METERS
CE SURFDATA 99999 1999          SURFNAME
CE UAIRDATA 99999 1999          UAIRNAME
C23456789012345678901234567890123456789012345678901234567890
C 99999 99 99999 99
C9 624 9 68.6439 0.7367 292.7 2 300.0 300.0
C9 62410 36.0762 3.5957 292.6 2 300.0 300.0
C
C          YEAR          MONTH          DAY
C          WRITE(9,1520)CF(L),CF(L)(6:7),CF(L)(9:10),CF(L)(12:13),
C          HOUR          DIRECT          SPEED          TEMP          STABILITY
C          1  CF(L)(15:16),WDIR(L),WVEL(L),AIRT(L),STABCL(L),
C          MIXING HT          MIXING HT
C          1  MIXHT(1,L),MIXHT(1,L)
1520  FORMAT(A105,T111,4A2,2F9.4,F6.1,I2,2F7.1)
1515  CONTINUE
      ELSEIF (.NOT.UZAB)THEN !UNUSABLE RECORD, WRITE OUT BLANKS FOR RIGHT SIDE
        DO 91515 L=1,24
          WRITE(9,91520)CF(L),BLANK
91520  FORMAT(A105,A56)
91515  CONTINUE
      ENDIF
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C          STEP 6 - FINISH UP:  NORMALLY OR WITH ERROR MESSAGE
          END DO
C          END OF FILE REACHED IN MET FILE - STOP PROGRAM
997  STOP
C          ERROR OCCURED IN I/O - PRINT ERROR MESSAGE AND QUIT
998  WRITE (*,*) 'I/O ERROR WHILE ATTEMPTING TO READ MET DATA'
      WRITE (*,*) 'ERROR WAS IN SAME DAY AS RECORD', RECORD
      WRITE (*,*) 'PROGRAM ENDED DUE TO ERROR.'
      STOP
          END
C          Last change:  BJ 1 Oct 2007 10:06 am
          LOGICAL FUNCTION ASK(MESS,LEN)
C
C          TAKES MESSAGE IN ARRAY MESS OF LENGTH N AND ASKS YES NO QUESTION
C          OF FORM --ASK QUESTION?(Y/N)
C          THEN IT READS IN Y/N ANSWER AND SETS ITSELF TRUE OR FALSE
C
          IMPLICIT INTEGER (A-Z)
C          BYTE MESS(50),FORM(30),ANS
C          ENCODE (30,100,FORM)LEN

```

```

C100      FORMAT ('('$--',I2,'A1','?(Y/N) ')')
CD        WRITE (5,200)LEN,FORM
CD200     FORMAT (1X,'LEN=',I2,'FORM= ',30A1)
C
C PREVIOUS COMMENTED LINES LEFT IN FOR HISTORICAL PURPOSES
C AS THEY WAY IT WAS WITH ENCODE STATEMENTS ON THE PDP
      CHARACTER*50 MESS
      CHARACTER*30 FORM
      CHARACTER*1 ANS
C
C CREATE APPROPRIATE FORMAT
      WRITE (FORM,100)LEN
100      FORMAT(' (1X,'--',A',I2,'','?(Y/N) '\)')
c        write (0,150)form
c150     format(a30)
2        CONTINUE
      WRITE (6,FORM)MESS(1:LEN)
      READ (5,350)ANS
350      FORMAT (A1)
      IF (ANS.EQ.'Y'.OR.ANS.EQ.'y')ASK=.TRUE.
      IF (ANS.EQ.'N'.OR.ANS.EQ.'n')ASK=.FALSE.
      IF (ANS.EQ.'N'.OR.ANS.EQ.'Y'.OR.ANS.EQ.'n'.or.ANS.EQ.'y')RETURN
      WRITE (6,400)
400      FORMAT(1X,'TYPE A "Y" OR AN "N" TO ANSWER QUESTION ')
      GO TO 2
      END

SUBROUTINE OPENFL

      CHARACTER*80 COMLIN !12.2.96 COMMAND LINE FILENAME ARG
      COMMON /COMAND/COMLIN !COMMUNICATE WITH SUBROUTINE
      INTEGER INDEX !BRJ070927 added because it appears that the system handles
command line
      !strings differently than it used to. So, must figure out where
extent of
      !filename is in command line and use only that to open, the
portion after
      !filename (which is the station code, which is preceded by a
space)
      !is now part of what the open statement is trying to open
      CHARACTER*80 COMOUT !12.2.96
      INTEGER P1,P2,P3

      WRITE(6,1010)comlin(1:60)
1010     FORMAT(1x,'comlin= ',a60)
      P1=INDEX(COMLIN,'.')
      P2=INDEX(COMLIN,' ')
      P3=INDEX(COMLIN(2:80),' ')
      WRITE(6,1015)P1,P2,P3,COMLIN(1:P2-1)
1015     FORMAT(1X,3I5,A11)
      OPEN (UNIT=8, FILE=COMLIN(1:P2-1), STATUS='OLD')

C COMMAND LINE 2 ARGUMENTS, INPUT FILENAME AND STATION #
C 'CIMPRO 89KER54.DET 054'
C THE 6,7 POSITIONS ON INPUT FILENAME ARE NOT RELIABLE FOR STATION#
C GETCL KEEPS THE SPACE BETWEEN ARGUMENTS
C OUTPUT FILENAME WILL HAVE 8 CHARS, YYCCSSS.2PC

      COMOUT(1:5)=COMLIN(1:5)
      COMOUT(6:8)=COMLIN(13:15)
      COMOUT(9:12)='.2PC'
      OPEN( UNIT=9,FILE=COMOUT,STATUS='NEW')
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCC BRJ101116 CCCCCCCCCCCCCCCCCC
C
C ADDED CHECK FOR 'N' FLAG, THOUGH IN THEORY N MEANS NOT CALCULATED
C AND IS STATION SPECIFIC, SO ALL HOURS FOR THIS SENSOR WOULD BE FLAGGED
C AND CIMPRO1 WOULD EXCLUDE ALL DAYS, BUT MAKE VALIDA IN CIMPRO2 CONSISTENT
C WITH VALIDA IN CIMPRO1

```

```

C
C ALSO SEE DISCUSSION OF QC FLAGS IN TECHNICAL MANUAL (WHICH IS CURRENTLY IN PROGRESS)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC BRJ070927 CCCCCCCCCCCCCCCCCC
C
C FUNCTION VALID, CHANGED TO FUNCTION VALIDA, TO REFLECT THE
C NEW CIMIS QC CODES IN USE, 'R' IS AT USER'S DISCRETION
C
C FLAG INCLR IF TRUE, USER WISHES TO REGARD DATA FLAGGED WITH 'R' AS BAD
C
C
C
C
C
C
C _____
C
C CHECKQ LOOKS FOR QUALITY FLAGS WHICH ARE NOT '*' OR 'F'. IT REPLACES
C THESE BAD VALUES WITH THE AVERAGE OF THE TWO ADJOINING VALUES. A BAD
C VALUE IN HOUR 1 IS REPLACED WITH THE FIRST VALID VALUE, AND A BAD VALUE
C IN HOUR 24 IS REPLACED BY THE LAST VALID VALUE.
C
C '*' = NORMAL             ' ' = ALSO NORMAL
C 'C' = NOT COLLECTED     'S' = SENSOR OUT OF SERVICE
C 'M' = NOT AVAILABLE     'R' = OUT OF RANGE SEVERE
C 'F' = ESTIMATED        'Y' = OUT OF RANGE
C
C
C _____
C
C IS THE CHARACTER A VALID ONE ?
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      LOGICAL FUNCTION VALIDA (QFLAG,ERRMESS)
      CHARACTER*1 QFLAG
      CHARACTER*10 ERRMESS !BRJ070926 ADDED ERRMESS TO DEAL WITH WINDSPEED=1.0MPH
SITUATION
      LOGICAL INCLR           !BRJ070927
      COMMON /REJCRI/RC,INCLR      !BRJ070927
      INTEGER RC
      VALIDA=.TRUE. !INNOCENT UNTIL PROVEN GUILTY THIS TIME
C      IF((QFLAG.EQ.' ').OR.(QFLAG.EQ.*').OR.(QFLAG.EQ.'F'))THEN
C         VALID = .TRUE.
C      ELSE
C         VALID = .FALSE.
C      END IF
c      WRITE(6,6060)ERRMESS
c6060      FORMAT(1X,'VALIDA: ERRMESS= ',A10)
          !BRJ101116 ADDED 'N' TO LIST OF AUTOMATICALLY BAD VALUES
          IF(QFLAG.EQ.'M'.OR.QFLAG.EQ.'S'.OR.QFLAG.EQ.'N')THEN !INCLUDING M,S,N HERE THOUGH
N          PROBABLY MEANS ALL 24H BAD AND DAY FLAGGED AS UNUSABLE BY CIMPRO1
          VALIDA=.FALSE.
          RETURN
          ELSEIF(QFLAG.EQ.'I')THEN
          IF (ERRMESS(1:7).EQ.'WINDVEL') THEN !IF THIS IS WINDSPEED, THEN ASSUME VALID
TRUE          DUE TO 1MPH SPEED.
          VALIDA=.TRUE.
          RETURN
          ELSE
          VALIDA=.FALSE.
          RETURN
          ENDIF
          ELSEIF(QFLAG.EQ.'R')THEN           !BRJ070927
          IF (INCLR) THEN !THE USER WANTS TO CLASSIFY R AS A BAD DATA VALUE
          VALIDA=.FALSE.
          ELSE
          VALIDA=.TRUE. !IGNORE R
          ENDIF
          ENDIF
          RETURN
          END
C
C _____

```

```

C     RETURNS THE NUMBER OF THE NEXT VALID CHAR, STARTING AT PASSED INDEXP
      INTEGER FUNCTION NEXTV (ARRAYQ, INDEXP,ERRMESS) !BRJ070927

      INTEGER INDEXP
      CHARACTER ARRAYQ
      DIMENSION ARRAYQ(24)
      LOGICAL VALIDA !MUST DEFINE TYPE OF FUNCTION BEFORE USE
      CHARACTER*10 ERRMESS !BRJ070927

      NEXTV=0
c     WRITE(6,10)ERRMESS(1:10)
c100  FORMAT(1X,'NEXTV: ERRMESS= ',A10)
      IF (INDEXP .EQ. 24) RETURN !SINCE OUT OF RANGE

      DO I=INDEXP,24
      IF (VALIDA(ARRAYQ(I),ERRMESS)) THEN
      NEXTV=I
      RETURN
      END IF
      END DO

      RETURN
      END

C     _____

C     RETURNS THE NUMBER OF THE PREVIOUS VALID CHAR, STARTING AT PASSED INDEXP
      INTEGER FUNCTION PREV (ARRAYQ, INDEXP,ERRMESS) !BRJ070927

      INTEGER INDEXP
      CHARACTER ARRAYQ
      DIMENSION ARRAYQ(24)
      LOGICAL VALIDA !BRJ070927
      CHARACTER*10 ERRMESS !BRJ070927

      PREV=0

      IF (INDEXP .EQ. 1) RETURN !SINCE OUT OF RANGE

      DO I=INDEXP,1,-1
      IF (VALIDA(ARRAYQ(I),ERRMESS)) THEN
      PREV=I
      RETURN
      END IF
      END DO

      RETURN
      END

C     _____

      SUBROUTINE CHECKQ (ARRAY, ARRAYQ, RECORD, DEBUG2,ERRMESS)
CCCCCCCCCCCCCCCCCCCC BRJ070925 CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C NOTE, I DID NOT CHANGE THE NAME OF THIS SUBORUTINE TO CHECKQA, AS IN CIMPRO1V2 BECAUSE
C THIS SUBROUTINE IS SUBSTANTIALLY DIFFERENT. I ADDED ARGUMENT ERRMESS, TO TIP OFF
C VALIDA AS TO WHETEHR OR NOT IT'S DEALING WITH A WIND SPEED VALUE
C ADDED 'ERRMESS' ARGUMENT TO TIP OFF VALIDA
C
C I HAVE CHANGED THE QAQC STUFF SOME. (1) CIMIS CHANGED THEIR FLAGS SOME
C (2) I'M LOOSENING UP THE RESTRICTIONS SOME (3) USER DETERMINES IF
C 'R' IS BAD DATA OR NOT INCLR IS FLAG TRUE --> YES 'R' IS BAD
C 'R' IN THE LIST OF FLAGS TO WORRY ABOUT
C
C CURRENT HOURLY SENSOR FLAG SUMMARY (FROM CIMIS WEBSITE 9/26/07)
C

```

```

C SEVERE FLAGS
C 'I' = DATA ALUE HAS NO MEANING/IGNORE (NOTE: FOR 2006 PARLIER, THIS WAS ONLY USED
C                                     FOR WIND SPEEDS AT 1.0MPH PROBABLY BECAUSE
C                                     THOSE DEVICES DON'T REPORT ANYTHING LESS.
C                                     I WILL HANDLE THIS CASE BY IGNORING IT AND
C                                     USING THE 1MPH - THESE GENERALLY CLASSED AS
C                                     CALMS ANYWAY)
C 'M' = DATA VALUE MISSING
C 'S' = SENSOR OUT OF SERVICE OR DATA OUT OF SENSOR THRESHOLD
C
C INFORMATIVE FLAGS
C 'A' = HISTORICAL AVG
C 'E', 'T' = HISTORICAL AVG USED TO CALCULATE VALUE (NOTE: I HAVE ALSO USED AN 'E' TO
INDICATE
C                                     THAT I INTERPOLATED A VALUE)
C 'N' = DATA VALUE NOT COLLECTED BY THIS STATION
C 'P' = QUALITY TEST PENDING
C 'Q' = QC COULD NOT BE PERFORMED
C 'R' = DATA FAR OUT OF HISTORICAL LIMITS (NOTE: THIS FLAG IS OPTIONAL, USER CAN
C                                     DECIDE WHETHER IT'S BAD VALUE OR NOT)
C 'Y' = DATA MODERATELY OUT OF HISTORICAL LIMITS
C
C MY INTENTION IS TO USE I M S FLAGS AND CALL THOSE DATA INVALID, THE REST WILL BE VALID.
C
C END OF BRJ070925 COMMENTS CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
REAL ARRAY, DIFFER
CHARACTER ARRAYQ
DIMENSION ARRAY(24), ARRAYQ(24)
INTEGER NEXT, PREV, NEXTV, PREVV !DEFINE FUNCTION TYPE ALSO
LOGICAL VALIDA !MUST DEFINE TYPE OF FUNCTION BEFORE USE
INTEGER RECORD !FOR ERROR CHECKING
LOGICAL DEBUG2
CHARACTER*10 ERRMESS !USED TO TIP OFF VALIDA BRJ070927
COMMON/REJCR/RC,INCLR !BRJ070927 USED TO TELL VALIDA WHETHER 'R' IS BAD OR NOT
INTEGER RC
LOGICAL INCLR

C CHECK TO SEE IF WHOLE DAY IS BAD. IF SO, WRITE OUT ERROR AND QUIT.
NEXT = NEXTV(ARRAYQ, 1, ERRMESS)
IF (NEXT .EQ. 0) THEN
WRITE (*,*) 'ERROR: AN ENTIRE DAY OF MET DATA IS BAD.'
WRITE (*,*) 'ERROR WAS IN SAME DAY AS RECORD', RECORD
WRITE (*,*)
1'YOU MUST CORRECT AT LEAST ONE HOUR, AND RERUN THE PROGRAM.'
STOP
END IF

C MAKE SURE THE FIRST ELEMENT IS VALID
IF (.NOT. VALIDA(ARRAYQ(1),ERRMESS)) THEN
ARRAY(1) = ARRAY(NEXT)
END IF
ARRAYQ(1)= ' '

C MAKE SURE THE LAST ELEMENT IS VALID
IF (.NOT. VALIDA(ARRAYQ(24),ERRMESS)) THEN
ARRAY(24) = ARRAY(PREVV(ARRAYQ, 24,ERRMESS))
END IF
ARRAYQ(24)= ' '

C {? DEBUG ?}
IF (DEBUG2) THEN
WRITE (*,*) '/////////////////////////////////////'
WRITE (*,*) 'CHECK Q DUMP'
END IF

C NOW LOOP THROUGH ALL OTHER ELEMENTS OF THE ARRAY
DO I=2,23
IF (.NOT. VALIDA(ARRAYQ(I),ERRMESS)) THEN

C SPLIT DIFFERENCE OVER MULTIPLE BLANK VALUES

```



```

PREV=PREVV(ARRAYQ, I,ERRMESS)
NEXT=NEXTV(ARRAYQ, I,ERRMESS)
DIFFER = (ARRAY(NEXT) - ARRAY(PREV))/(NEXT-PREV)
ARRAY(I) = ARRAY(I-1) + DIFFER

IF (DEBUG2) THEN
WRITE (*,*) I, ' ', ARRAY(I), ' ', PREV, ' ', NEXT,
1 ' ', DIFFER
END IF

END IF
END DO

END

C
C
C
C
C DOSTAB CALCULATED THE STABILITY CLASS (1-6) BASED ON THE STANDARD
C DEVIATION OF THE WIND DIRECTION, SIGMAT

C FOR MORE INFORMATION, SEE AIR POLLUTION MODELING, ZANNETTI (1990)
C PP 148-149. OR SEE ORIGINAL REFERENCE:
C US EPA (1986): GUIDELINE ON AIR QUALITY MODELS. EPA DOCUMENT
C EPA-450/2-78-025. RESEARCH TRIANGLE PARK, NORTH CAROLINA.
C
C DOSTAB REVISED 2/2/00 TO REFLECT FOLLOWING REFERENCE (BJ)
C
C UPDATED REFERENCE "ON-SITE METEOROLOGICAL PROGRAM GUIDANCE FOR
C REGULATORY MODELING APPLICATIONS" RELEASED 3/15/99 AVAILABLE ON
C SCRAM BULLETIN BOARD OF EPA, (DOCUMENT IS STILL DATED JUNE 1987,
C BUT CONTAINS UPDATED SECTIONS)
C
C NOTE TO MYSELF: 10/2/00 WVLEL CONTAINS 0 WIND SPEEDS FOR THOSE
C BELOW THE THRESHOLD. SINCE THE THRESHOLD IS BELOW ANY OF THE CUTOFF
C POINTS, THIS DOES NOT MAKE A DIFFERENCE, BUT KEEP IN MIND.....

SUBROUTINE DOSTAB (STABCL, NETRAD, WVLEL, SIGMAT, OSTAB, DEBUG3)

INTEGER STABCL, OSTAB
REAL NETRAD, WVLEL, SIGMAT, MAXDEG
LOGICAL DEBUG3

DIMENSION MAXDEG(6)

C MAXIMUM SIGMAT IN DEGREES FOR EACH CLASS (A REALLY > 22.5)
C
C A B C D E F
DATA MAXDEG /360., 18.0, 16.2, 11.9, 7.9, 5.1 / !THIS IS ADJUSTED TO CIMIS
CONDITIONS
C THE ABOVE LINE REPRESENTS 2 ADJUSTEMENTS OFF OF THE CONVENTIOANL VALUES
C THESE ADJUSTEMENTS ARE OUTLINED IN ON-SITE METEOROLOGICAL PROGRAM GUIDANCE
C FROM EPA ON P 6-31, ONE ADJUSTEMENT FOR 3 CM ROUGHNESS HEIGHT (GRASS ) AND
C THE OTHER FOR 2M MEASUREMENT HEIGHT, THE 3 CM COMES FROM EPA TABLE IN
C ONSITE GUIDE
C COMPUTE DAYTIME STABILITY CLASS
DO I=1,6
IF (SIGMAT .LT. MAXDEG(I)) STABCL = I
END DO

IF (DEBUG3) THEN
WRITE (*,*) '-----'
WRITE (*,*) 'DAYTIME SIGMAT: ', STABCL, ' ', SIGMAT
END IF

C NIGHTTIME IS ASSUMED TO BE WHEN NETRAD <= 0.
C DAYTIME IS WHEN NETRAD > 0.

```

```

IF (NETRAD .GT. 0.) THEN !DAYTIME CASE

  IF (STABCL .EQ. 1) THEN !INITIAL STABILITY WAS A OR 1

    IF (WVEL .LT. 3.0) THEN
      STABCL = 1 !A
    ELSE IF (WVEL .LT. 4.0) THEN
      STABCL = 2 !B
    ELSE IF (WVEL .LT. 6.0) THEN
      STABCL = 3 !C
    ELSE IF (WVEL .GE. 6.0) THEN
      STABCL = 4 !D
    END IF

  ELSE IF (STABCL .EQ. 2) THEN !INITIAL STAB ESTIMATE WAS B OR 2

    IF (WVEL .LT. 4.0) THEN
      STABCL = 2 !B
    ELSE IF (WVEL .LT. 6.0) THEN
      STABCL = 3 !C
    ELSE IF (WVEL .GE. 6.0) THEN
      STABCL = 4 !D
    END IF

  ELSE IF (STABCL .EQ. 3) THEN !INITIAL STAB EST WAS C OR 3

    IF (WVEL .LT. 6.0) THEN
      STABCL = 3 !C
    ELSE IF (WVEL .GE. 6.0) THEN
      STABCL = 4 !D
    END IF

  ELSE IF (STABCL .GE. 4) THEN !INITIAL STAB EST WAS D,E,F
    STABCL = 4

  END IF

ELSE IF (NETRAD .LT. 0) THEN !NIGHTTIME SITUATION

  IF (STABCL .EQ. 1) THEN !INITIAL STAB EST WAS A

    IF (WVEL .LT. 2.9) THEN
      STABCL = 6 !F
    ELSE IF (WVEL .LT. 3.6) THEN
      STABCL = 5 !E
    ELSE IF (WVEL .GE. 3.6) THEN
      STABCL = 4 !D
    END IF

  ELSE IF (STABCL .EQ. 2) THEN !INITIAL STAB EST WAS B

    IF (WVEL .LT. 2.4) THEN
      STABCL = 6 !F
    ELSE IF (WVEL .LT. 3.0) THEN
      STABCL = 5 !E
    ELSE IF (WVEL .GE. 3.0) THEN
      STABCL = 4 !D
    END IF

  ELSE IF (STABCL .EQ. 3) THEN !INITIAL STAB EST WAS C

    IF (WVEL .LT. 2.4) THEN
      STABCL = 5 !E
    ELSE IF (WVEL .GE. 2.4) THEN
      STABCL = 4 !D
    END IF

  ELSE IF (STABCL .EQ. 4) THEN !INITIAL STAB EST WAS D
    STABCL = 4 !D

```

```

ELSE IF (STABCL .EQ. 5) THEN !INITIAL STAB EST WAS E
  IF (WVEL .LT. 5.0) THEN
    STABCL = 5 !E
  ELSE IF (WVEL .GE. 5.0) THEN
    STABCL = 4 !D THIS WAS ERRONEOUSLY SET AT F IN V3, FIXED IN V4
120806BRJ
  END IF

ELSE IF (STABCL .EQ. 6) THEN ! INITIAL STAB EST WAS F
  IF (WVEL .LT. 3.0) THEN
    STABCL = 6 !F
  ELSE IF (WVEL .LT. 5.0) THEN
    STABCL = 5 !E
  ELSE IF (WVEL .GE. 5.0) THEN
    STABCL = 4 !D
  END IF

  END IF !END OF NIGHT TIME CASES
END IF ! END OF DAY VS NIGHT CONSTRUCT

IF (DEBUG3) THEN
WRITE (*,*) 'NIGHTTIME SIGMAT: ', STABCL, ' ', WVEL
END IF

C NOW, ADJUST STABILITY CLASS SO THAT IT CAN ONLY VARY BY
C ONE CLASS PER HOUR. IGNORE FIRST TIME WHEN OSTAB=0.

IF ((ABS(OSTAB - STABCL) .GT. 1) .AND. (OSTAB .NE. 0)) THEN
IF ( (STABCL - OSTAB) .GT. 0) THEN
  STABCL = OSTAB + 1 !INCREASE CLASS BY 1
ELSE
  STABCL = OSTAB - 1 !DECREASE CLASS BY 1
END IF
END IF

IF (DEBUG3) THEN
WRITE (*,*) 'ADJUSTED SIGMAT: ', STABCL, ' ', OSTAB
END IF

OSTAB = STABCL
RETURN
END

SUBROUTINE MKNM(STA,JD,YR,ONAME)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CREATES 8 CHARACTER NAME OUT OF 3 INTEGERS
C STA=STATION CODE (I3)
C JD=JULIAN DAY (I3)
C YR=YEAR (I2)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
IMPLICIT INTEGER(A-Z)
CHARACTER*8 NAME,ONAME
CHARACTER*3 CSTA,CJD
CHARACTER*2 CYR
EQUIVALENCE (CSTA(1:3),NAME(1:3)),(CJD(1:3),NAME(4:6)),
1 (CYR(1:2),NAME(7:8))

WRITE(CSTA,100)STA
100 FORMAT(I3)
WRITE(CJD,100)JD
WRITE(CYR,200)YR
200 FORMAT(I2)
DO 10 I=1,2
  IF(CJD(I:I).EQ.' ')CJD(I:I)='0'
  IF(CSTA(I:I).EQ.' ')CSTA(I:I)='0'

```

```

10      CONTINUE
        ONAME(1:8)=NAME(1:8)
        RETURN
        END

        SUBROUTINE FIXER(CF,NETRAD,AIRT,WVEL,WDIR,
1SIGMAT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C THIS SUBROUTINE REPLACES A MISSING VALUE AS DEFINED BY VALID (TRUE OK, FALSE, MISSING)
C WITH THE INTERPOLATED VALUE FOR DEBUGGING AND CHECKING PURPOSES AND FLAGS THE
C REPLACEMENT WITH AN '@' IN THE QUALITY CONTROL COLUMN
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        IMPLICIT INTEGER(A-Z)
        REAL NETRAD,AIRT,WVEL,WDIR,SIGMAT
        REAL TEMP
        CHARACTER*12 FORM
        CHARACTER*105 CF
        LOGICAL VALIDA
        CHARACTER*10 ERRMESS !BRJ070927

C016,92-01-01,01, 1,*, -3.306,*, -69.174,*, 45.860,*, 8.083,*, 134.300,*,
5.778,*, 0.000#
        ERRMESS(1:10)='XXXXXXXXXX' !BRJ070927
        IF(.NOT.VALIDA(CF(34:34),ERRMESS))THEN !NETRAD
            TEMP=NETRAD*2.065 !CONVERT FROM W/M2 BACK TO LANGLEYS BRJ070927
            WRITE(FORM,100)TEMP
100      FORMAT(T3,F9.3)
            CALL FIX2(34,CF,FORM)
        ENDIF
        IF(.NOT.VALIDA(CF(46:46),ERRMESS))THEN !AIRT BRJ070927
            TEMP=(9.*(AIRT-273.15)/5.) + 32. !CONVERT BACK TO FAHRENHEIT FROM KELVIN
            WRITE(FORM,100)TEMP
            CALL FIX2(46,CF,FORM)
        ENDIF
        ERRMESS(1:10)='WINDVEL ' !BRJ070927
        IF(.NOT.VALIDA(CF(58:58),ERRMESS))THEN !WVEL BRJ070927
            TEMP=WVEL*2.237 !CONVERT BACK TO MPH
            WRITE(FORM,100)TEMP
            CALL FIX2(58,CF,FORM)
        ENDIF
        ERRMESS(1:10)='XXXXXXXXXX' !BRJ070927
        IF(.NOT.VALIDA(CF(70:70),ERRMESS))THEN !WDIR !BRJ070927
            TEMP=AMOD(180.+WDIR,360.)
            WRITE(FORM,100)TEMP
            CALL FIX2(70,CF,FORM)
        ENDIF
        IF(.NOT.VALIDA(CF(82:82),ERRMESS))THEN !SIGMAT
            WRITE(FORM,100)SIGMAT
            CALL FIX2(82,CF,FORM)
        ENDIF
        RETURN
        END

        SUBROUTINE FIX2(N,CF,FORM)
        IMPLICIT INTEGER(A-Z)
        CHARACTER*105 CF
        CHARACTER*12 FORM

        FORM(2:2)=' '
        FORM(12:12)=' '
        FORM(1:1)='@'
        CF(N:N+11)=FORM(1:12)
        RETURN
        END

        SUBROUTINE IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C

```

```

C CALLING PROGRAM NEEDS TO DECLARE THE FOLLOWING
C-----
C CHARACTER*35 FIN,FOUT
C LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
C INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
C UIN=1 OR WHATEVER IS APPROPRIATE TO FIT INTO CALLING ROUTINE
C UOUT=2
C-----
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER UIN,UOUT
      CHARACTER*35 FIN,FOUT
      LOGICAL INSTATE,OUTSTATE,INSTSTATUS
      CHARACTER*70 CLINE

      CALL GETCL(CLINE)
      CALL IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)

      IF(INSTATE.AND.OUTSTATE)THEN
C          WRITE(6,500)FIN,FOUT,K1,K2
C500      FORMAT(1X,'IOREADY: INPUT,OUTPUT ',/1X,A35,/1X,A35,
C          * /1X,'K1,K2 ',2I3)
          CONTINUE !STUFF ABOVE FOR DEBUGGING
      ELSEIF (.NOT.INSTATE)THEN
          WRITE(6,505)FIN,K1
C505      FORMAT(1X,'IOREADY: INPUT FILENAME ERROR ',A35,1X,I3)
          STOP
      ELSEIF (.NOT.OUTSTATE)THEN
          WRITE(6,510)FOUT,K2
C510      FORMAT(1X,'IOREADY: OUTPUT FILENAME ERROR ',A35,1X,I3)
          STOP
      ENDIF
      !CHECK TO MAKE SURE THE INPUT FILE EXISTS
      INQUIRE(FILE=FIN,EXIST=INSTSTATUS)
      IF(.NOT.INSTSTATUS)THEN
C255      WRITE(6,255)FIN
          *   FORMAT(1X,'IOREADY: INPUT FILE NOT EXIST',
          *   /1X,A35, /1X,'PROGRAM STOPPING ')
          STOP
      ENDIF
      OPEN(UNIT=UIN,STATUS='OLD',FILE=FIN,ERR=1000)
      OPEN(UNIT=UOUT,STATUS='UNKNOWN',FILE=FOUT,ERR=2000)
      RETURN

      !ERRORS ON OPENING GO BELOW
C1000     CONTINUE
          WRITE(6,1100)FIN
C1100     FORMAT(1X,'ERROR TRYING TO OPEN IN FILE ',/1X,A35)
          STOP
C2000     CONTINUE
          WRITE(6,2100)FOUT
C2100     FORMAT(1X,'ERROR TRYING TO OPEN OUT FILE ',/1X,A35)
          STOP
      END SUBROUTINE

      SUBROUTINE IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)
      IMPLICIT NONE
      CHARACTER*35 FIN,FOUT
      CHARACTER*70 CLINE
      INTEGER K1,K2,IBLANK,K,I,L
      LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
      !K1, K2 ARE LENGTHS OF THE FILENAMES

      DO I=1,35
          FIN(I:I)=' '
          FOUT(I:I)=' '
      END DO
      !ASSUME CLINE HAS 1 OR MORE BLANK SPACES WHICH DIVIDE THE TWO FILENAMES
      !ASSUME THAT INPUT FILE IS THE FIRST FILE, AND OUTPUTFILE IS THE SECOND ONE
      K=LEN_TRIM(CLINE) !FIND OUT TOTAL STRING LENGTH
      !LOOK FOR FIRST BLANK

```

```

IBLANK=0
DO I=1,K
  IF (CLINE(I:I).EQ.' ')THEN
    IBLANK=I
    GOTO100
  ELSE
    CONTINUE
  ENDIF
END DO
100  IF (IBLANK.EQ.0) THEN
      INSTATE=.FALSE.
      OUTSTATE=.FALSE.
      K1=0
      K2=0
      WRITE(6,200)CLINE
200  FORMAT(1X,'COMMAND LINE LACKING 2 FILENAMES ',/1X,A70)
      RETURN
    ELSEIF (IBLANK.GE.70)THEN
      WRITE(6,250)CLINE
250  FORMAT(1X,'FILENAME TOO LONG ',/1X,A70)
      K1=IBLANK
      K2=0
      INSTATE=.TRUE.
      OUTSTATE=.FALSE.
    ELSE
      FIN(1:IBLANK-1)=CLINE(1:IBLANK-1)
      L=K-(IBLANK-1)
      FOUT(1:L)=CLINE(IBLANK:K)
      FOUT(1:L)=ADJUSTL(FOUT(1:L))
      K2=LEN_TRIM(FOUT(1:L))
      IF(K2.EQ.0)THEN
        INSTATE=.TRUE.
        OUTSTATE=.FALSE.
        RETURN
      ENDIF
      K1=IBLANK-1
      INSTATE=.TRUE.
      OUTSTATE=.TRUE.
    ENDIF
  RETURN
END SUBROUTINE

```

```

      real Function FIXNETRAD(M,H,NR)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C BRJ 12/12/2013
C THIS FUNCTION WAS CREATED TO FIX SITUATIONS WHERE THE NET RADIATION
C DURING DAYTIME HOURS IS NEGATIVE. TEMESGEN NOT SURE WHY
C THIS HAPPENS.
C GENERALLY NO ERROR FLAGS WITH DAYTIME NEGATIVE NET RAD
C SO STABILITY CLASS CALCULATOR (DOSTAB)
C ASSIGNS NIGHTTIME STABILITIES TO DAYTIME HOURS
C
C TABLE CREATED BASED ON SUNRISE/SUNSET HOURS FOR REDDING AND SAN DIEGO FOR 2013
C USED REDDING SUNRISE AND SD SUNSET, ADDED 1 HOUR SUNRISE, SUBTRACTED
C 1 HOUR FOR SUNSET
C NOTE THAT USEPA (2000 METEOROLOGICAL MONITORING GUIDANCE PG6-12 USES
C 1 HOUR AFTER SUNRISE AND 1 HOUR
C BEFORE SUNSET AS NIGHT TIME), THIS TABLE APPROXIMATES THAT
C BECAUSE THE HOUR SPECIFIED WILL BE DEFINED AS NIGHT TIME
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT NONE
      INTEGER M,H
      REAL NR
      INTEGER LASTHOURNIGHT(12),FIRSTHOURNIGHT(12)
      DATA LASTHOURNIGHT/9,8,7,6,6,6,6,6,7,7,8,8/
      DATA FIRSTHOURNIGHT/16,17,17,17,18,18,18,18,17,16,16,16/

      FIXNETRAD=NR

```

```
      IF (H.LE.LASTHOURNIGHT(M).OR.H.GE.FIRSTHOURNIGHT(M)) THEN !IT'S NIGHT, DON'T NEED
TO DO ANYTHING
      RETURN
    ELSE
      IF (NR.LE.0.) THEN
        FIXNETRAD=ABS(NR)+0.01
      ELSE
        RETURN
      ENDIF
    ENDIF
  RETURN
ENDDC      Last change:  BJ  16 Nov 2010   3:01 pm
```

MAKISCV2.FOR

```
C      Last change:  BJ   25 Oct 2010   3:48 pm
      PROGRAM MAKISCV2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C[101025]BRJ CALL THE IOREADY SUBROUTINE, ALSO GET THE
C CIMIS STATION NUMBER AND YEAR FROM THE FIRST LINE OF THE
C FILE, INSTEAD OF FROM THE INPUT FILENAMES, THE COMMAND LINE
C IS CHANGED TO MAKISCV2 INFILE OUTFILE
C THIS WILL MAKE THE QUADRAD OF PROGRAMS: PREFORM, CIMPRO1V3, CIMPRO2V5 AND
C MAKISCV2 THE SAME IN TERMS OF RUNNING WITH PROGRAMNAME INFILE OUTFILE AS THE
C SYNTAX FOR RUNNING, WORKING THESE CHANGES IN CONJUNCTION WITH EDGAR VIDREO.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C [090310]BRJ CHANGED CONSOLE READ/WRITE UNITS TO
C 5/6, RESPECTIVELY, PREVIOUSLY WAS '0', FROM THE
C ANCIENT DAYS
C
C GET COMMAND LINE FILE, PARSE FILENAME,
C ADD EXTENSION ISC, GRAB COLUMNS 111-158 AND
C PUT INTO FILE FOR USE WITH ISCST3, PROMPT
C USER TO ADD SURFACE/UPPER AIR STATIONS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT NONE
      CHARACTER*50 FIRSTLIN  !FIRST LINE OF MET FILE
C      CHARACTER*50 FIN      !COMMAND STRING (FILENAME) BRJ101025 COMMENTED OUT, REPLACED
BELOW FOR IOREADY
      CHARACTER*50 ISCOUT   !OUTPUT FILENAME CONTAIN ISC READY OUTPUT

C EXPECTING COMMAND LINE FILENAME OF FORM YYCCSSS.2P8
C CCC=3 CHAR COUNTY CODE, SSS=3 DIGIT STATION CODE FOR CIMIS

      CHARACTER*3 CCIM,SSS
      CHARACTER*2 YY
      CHARACTER*1 DUM

      CHARACTER*158 BIGLIN(24) !THIS READSIN THE LONG LINE FROM INPUT FILE
      CHARACTER*48 LINE(24)   !THIS RECEIVES ISCSPTION OF BIGLIN

      INTEGER I,J,K,H

      !IOREADY VARIABLES BRJ101025
      CHARACTER*35 FIN,FOUT
      LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
      INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
      UIN=1 !OR WHATEVER IS APPROPRIATE TO FIT INTO CALLING ROUTINE
      UOUT=2

C      CALL MAKM1 !COMMENTED OUT BRJ101025, THIS WILL BE ADDRESSED IN DOCUMENTATION
C      WRITE(6,7)
C7     FORMAT(1X,/,,'-----PRESS RETURN TO CONTINUE -----'/)
C      READ(5,8)DUM
C8     FORMAT(A1)

C      CALL GETCL(FIN) !GET COMMAND LINE FILENAME yycccSSS.2P8

C      YY(1:2)=FIN(1:2) !THIS IS THE YEAR !COMMENTED OUT, THESE COME FROM 1ST RECORD
NOW BRJ101025
C      SSS(1:3)=FIN(6:8) !STATION NUMBER !COMMENTED OUT, THESE COME FROM 1ST RECORD
NOW BRJ101025
      CALL IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT) !BRJ101025 ADDED THIS CALLAND
SBURROUTINE
C OPEN INPUT FILE
C      OPEN(UNIT=1,STATUS='OLD', FILE='FIN',ERR=1000)
C      OPEN(UNIT=1,STATUS='OLD', FILE=FIN) !NOT NEEDED HERE NOW BRJ101025

C CREATE OUTPUT FILENAME AND OPEN IT
```



```

C      ISCOUT(1:9)=FIN(1:9)  !COMMENTED OUT BRJ101025
C      ISCOUT(10:12)='ISC'  !COMMENTED OUT BRJ101025

C      OPEN(UNIT=2,STATUS='UNKNOWN',FILE=ISCOUT,ERR=2000)  COMMENTED OUT BRJ101025

      DO 117 I=1,50
117    FIRSTLIN(I:I)=' '
C      CREATE FIRST LINE OF ISC FILE, READ FIRST LINE OF INPUT FILE AND GET
C      STATION NUMBER AND YEAR BRJ101025
      READ(1,200)BIGLIN(1) !GET FIRST MET DATA LINE IN ORDER TO GET STATION AND YEAR
      REWIND (1) !AND REWIND IT IN PREPARATION FOR THE ACTUAL PROCESSING DOWN BELOW
C      HERE ARE THE FIRST FEW CHARACTERS OF 3 LINES FROM THE INPUT FILE
C      THE CIMIS STATION ID CODE IS 2:4 AND THE YEAR IS 6:7
C12345678901234567890
C0080,05-01-01,01,001,*,    0.000,
C0080,05-01-01,02,001,*,    0.000,
C0080,05-01-01,03,001,*,    0.000,
C      MUST USE NUMERICS IN STATION NAMES
C      WILL USE '11SSS' AND '11111' FOR SURFACE AIR AND UPPPER AIR
C      FIRSTLIN(3:4)='CI'
      SSS(1:3)=BIGLIN(1)(2:4)  !BRJ101025
      YY(1:2)=BIGLIN(1)(6:7)  !BRJ101025
      FIRSTLIN(3:4)='11'
      FIRSTLIN(5:7)=SSS(1:3)
      FIRSTLIN(10:11)=YY(1:2)
      FIRSTLIN(15:20)='11111'
      FIRSTLIN(24:25)=YY(1:2)
C      FIRST LINE HAS BEEN CREATED
C      WRITE IT OUT
      WRITE(2,13)FIRSTLIN(1:25)
13    FORMAT(A25)

C      READY FOR READ LOOP DAY BY DAY
1    CONTINUE
      DO 10 I=1,24
      READ(1,200,END=3000)BIGLIN(I)
200  FORMAT(A158)
10   CONTINUE

C      FIRST QUESTION, IS THIS A USABLE DAY?
      IF(BIGLIN(1)(105:105).EQ. '#')GOTO1  !READ NEXTDAY IF THIS DAY UNUSABLE

C      OK, NOW TRANSFER ISCST3 MET PORTION TO ARRAY LINE
C      AND CHECK TO MAKE SURE THAT HOURS GO IN SEQUENCE
      DO 11 K=1,24
      LINE(K)(1:48)=BIGLIN(K)(111:158)
      READ(BIGLIN(K)(15:16),17)H
17   FORMAT(I2)
      IF(H.NE.K)THEN  !ARE HOURS IN SEQUENCE?
      WRITE(6,7771)H,K
7771  FORMAT(1X,'MISMATCH: H= ',I3,' WHILE K= ',I3)
      WRITE(6,1717)BIGLIN(K),LINE(K)
1717  FORMAT(1X,'METDAY02: HOUR MISMATCH ',/1X,A158,/1X,A48)
      STOP
      ENDIF
11   CONTINUE

      DO 250 I=1,24
      WRITE(2,240)LINE(I)
240  FORMAT(A48)
250  CONTINUE
      GOTO1

1000 CONTINUE  !OPEN ERROR FOR INPUT FILE
      WRITE(6,1100)FIN
1100  FORMAT(1X,'ERROR TRYING TO OPEN ',A50)
      STOP
2000 CONTINUE  !OPEN ERROR FOR OUTPUT FILE
      WRITE(6,2200)ISCOUT
2200  FORMAT(1X,'ERROR TRYING TO OPEN OUTPUT FILE ',A50)

```

```

        STOP
3000    CONTINUE
        WRITE(6,3100)
3100    FORMAT(1X,'NORMAL TERMINATION ')
        CLOSE(1)
        CLOSE(2)
        STOP
        END
        SUBROUTINE makml
C
        WRITE(6,1909)
1909    FORMAT(1X,' PROGRAM MAKISC',
*/1X,' ',
*/1X,' COMMAND LINE: "MAKISC YYCCSSS.2P8"',
*/1X,' where YYCCSSS.2P8 is the processed CIMIS data',
*/1X,' file with',
*/1X,' ISC portion appended in columns 111-158 of eac',
*/1X,' h record and',
*/1X,' # in column 105 indicates nonusable hour of no',
*/1X,' usable day.')
        WRITE(6,1110)
1110    FORMAT(1X,' YY=year, CCC=3 char county code, SSS=3 digit C',
*/1X,' IMIS station #',
*/1X,' ',
*/1X,' BASIC IDEA: Program takes columns 111-158 for us',
*/1X,' able days, and',
*/1X,' creates ISC compatible met data file, includin',
*/1X,' g surface/upper',
*/1X,' air met station designators on first record')
        RETURN
        END

C      Last change:  BJ  25 Oct 2010   8:54 am
        SUBROUTINE IOREADY(INSTATE,OUTSTATE,UIN,UOUT,FIN,FOUT)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CALLING PROGRAM NEEDS TO DECLARE THE FOLLOWING
C-----
C CHARACTER*35 FIN,FOUT
C LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
C INTEGER UIN,UOUT !INPUT, OUTPUT UNIT NUMBERS
C UIN=1 OR WHATEVER IS APPROPRIATE TO FIT INTO CALLING ROUTINE
C UOUT=2
C-----
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        INTEGER UIN,UOUT
        CHARACTER*35 FIN,FOUT
        LOGICAL INSTATE,OUTSTATE,INSTATUS
        CHARACTER*70 CLINE

        CALL GETCL(CLINE)
        CALL IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)

        IF(INSTATE.AND.OUTSTATE)THEN
C          WRITE(6,500)FIN,FOUT,K1,K2
C500      FORMAT(1X,'IOREADY: INPUT,OUTPUT',/1X,A35,/1X,A35,
C          * /1X,'K1,K2 ',2I3)
        CONTINUE !STUFF ABOVE FOR DEBUGGING
        ELSEIF (.NOT.INSTATE)THEN
        WRITE(6,505)FIN,K1
505      FORMAT(1X,'IOREADY: INPUT FILENAME ERROR ',A35,1X,I3)
        STOP
        ELSEIF (.NOT.OUTSTATE)THEN
        WRITE(6,510)FOUT,K2
510      FORMAT(1X,'IOREADY: OUTPUT FILENAME ERROR ',A35,1X,I3)
        STOP
        ENDIF
        !CHECK TO MAKE SURE THE INPUT FILE EXISTS
        INQUIRE(FILE=FIN,EXIST=INSTATUS)
        IF(.NOT.INSTATUS)THEN

```

```

        WRITE(6,255)FIN
255      FORMAT(1X,'IOREADY: INPUT FILE NOT EXIST',
*        /1X,A35, /1X,'PROGRAM STOPPING ')
        STOP
        ENDIF
        OPEN(UNIT=UIN,STATUS='OLD',FILE=FIN,ERR=1000)
        OPEN(UNIT=UOUT,STATUS='UNKNOWN',FILE=FOUT,ERR=2000)
        RETURN

        !ERRORS ON OPENING GO BELOW
1000     CONTINUE
        WRITE(6,1100)FIN
1100     FORMAT(1X,'ERROR TRYING TO OPEN IN FILE ',/1X,A35)
        STOP
2000     CONTINUE
        WRITE(6,2100)FOUT
2100     FORMAT(1X,'ERROR TRYING TO OPEN OUT FILE ',/1X,A35)
        STOP
        END SUBROUTINE

        SUBROUTINE IOFILPRO(CLINE,INSTATE,FIN,K1,OUTSTATE,FOUT,K2)
        IMPLICIT NONE
        CHARACTER*35 FIN,FOUT
        CHARACTER*70 CLINE
        INTEGER K1,K2,IBLANK,K,I,L
        LOGICAL INSTATE,OUTSTATE !TRUE IF VALID NAME FOUND
        !K1, K2 ARE LENGTHS OF THE FILENAMES

        DO I=1,35
            FIN(I:I)=' '
            FOUT(I:I)=' '
        END DO
        !ASSUME CLINE HAS 1 OR MORE BLANK SPACES WHICH DIVIDE THE TWO FILENAMES
        !ASSUME THAT INPUT FILE IS THE FIRST FILE, AND OUTPUTFILE IS THE SECOND ONE
        K=LEN_TRIM(CLINE) !FIND OUT TOTAL STRING LENGTH
        !LOOK FOR FIRST BLANK
        IBLANK=0
        DO I=1,K
            IF(CLINE(I:I).EQ.' ')THEN
                IBLANK=I
                GOTO100
            ELSE
                CONTINUE
            ENDIF
        END DO
100      IF (IBLANK.EQ.0) THEN
            INSTATE=.FALSE.
            OUTSTATE=.FALSE.
            K1=0
            K2=0
            WRITE(6,200)CLINE
200      FORMAT(1X,'COMMAND LINE LACKING 2 FILENAMES ',/1X,A70)
            RETURN
        ELSEIF (IBLANK.GE.70)THEN
250      WRITE(6,250)CLINE
            FORMAT(1X,'FILENAME TOO LONG ',/1X,A70)
            K1=IBLANK
            K2=0
            INSTATE=.TRUE.
            OUTSTATE=.FALSE.
        ELSE
            FIN(1:IBLANK-1)=CLINE(1:IBLANK-1)
            L=K-(IBLANK-1)
            FOUT(1:L)=CLINE(IBLANK:K)
            FOUT(1:L)=ADJUSTL(FOUT(1:L))
            K2=LEN_TRIM(FOUT(1:L))
            IF(K2.EQ.0)THEN
                INSTATE=.TRUE.
                OUTSTATE=.FALSE.
                RETURN
            ENDIF
        ENDIF

```

```
K1=IBLANK-1
  INSTATE= .TRUE.
  OUTSTATE= .TRUE.
ENDIF
RETURN
END SUBROUTINE
```

APPENDIX 2: Manual versus CIMPRO2V5 determined stability classifications

Manual versus CIMPRO2V5 determined stability classifications

Number	File Name	Date	Hour	Manually Calculated Stability	CIMPRO2V5 Calculated Stability
1	08CIM194.2PC	2/21/2008	1	5	5
2	08CIM194.2PC	2/21/2008	2	4	4
3	08CIM194.2PC	2/21/2008	3	4	4
4	08CIM194.2PC	2/21/2008	4	5	5
5	08CIM194.2PC	2/21/2008	5	5	5
6	08CIM194.2PC	2/21/2008	6	4	4
7	08CIM194.2PC	2/21/2008	7	5	5
8	08CIM194.2PC	2/21/2008	8	4	4
9	08CIM194.2PC	2/21/2008	9	3	3
10	08CIM194.2PC	2/21/2008	10	4	4
11	08CIM194.2PC	2/21/2008	11	4	4
12	08CIM194.2PC	2/21/2008	12	4	4
13	08CIM194.2PC	2/21/2008	13	4	4
14	08CIM194.2PC	2/21/2008	14	4	4
15	08CIM194.2PC	2/21/2008	15	4	4
16	08CIM194.2PC	2/21/2008	16	4	4
17	08CIM194.2PC	2/21/2008	17	4	4
18	08CIM194.2PC	2/21/2008	18	4	4
19	08CIM194.2PC	2/21/2008	19	4	4
20	08CIM194.2PC	2/21/2008	20	4	4
21	08CIM194.2PC	2/21/2008	21	4	4
22	08CIM194.2PC	2/21/2008	22	4	4
23	08CIM194.2PC	2/21/2008	23	5	5
24	08CIM194.2PC	2/21/2008	24	6	6
25	06CIM145.2PC	10/23/2006	1	5	5
26	06CIM145.2PC	10/23/2006	2	6	6
27	06CIM145.2PC	10/23/2006	3	6	6
28	06CIM145.2PC	10/23/2006	4	6	6
29	06CIM145.2PC	10/23/2006	5	6	6
30	06CIM145.2PC	10/23/2006	6	6	6
31	06CIM145.2PC	10/23/2006	7	6	6
32	06CIM145.2PC	10/23/2006	8	5	5
33	06CIM145.2PC	10/23/2006	9	4	4
34	06CIM145.2PC	10/23/2006	10	3	3
35	06CIM145.2PC	10/23/2006	11	2	2
36	06CIM145.2PC	10/23/2006	12	1	1
37	06CIM145.2PC	10/23/2006	13	1	1
38	06CIM145.2PC	10/23/2006	14	1	1
39	06CIM145.2PC	10/23/2006	15	1	1
40	06CIM145.2PC	10/23/2006	16	1	1
41	06CIM145.2PC	10/23/2006	17	1	1
42	06CIM145.2PC	10/23/2006	18	2	2
43	06CIM145.2PC	10/23/2006	19	3	3
44	06CIM145.2PC	10/23/2006	20	4	4
45	06CIM145.2PC	10/23/2006	21	5	5
46	06CIM145.2PC	10/23/2006	22	6	6
47	06CIM145.2PC	10/23/2006	23	6	6
48	06CIM145.2PC	10/23/2006	24	6	6

49	08CIM182.2PC	9/28/2008	1	5	5
50	08CIM182.2PC	9/28/2008	2	4	4
51	08CIM182.2PC	9/28/2008	3	4	4
52	08CIM182.2PC	9/28/2008	4	5	5
53	08CIM182.2PC	9/28/2008	5	6	6
54	08CIM182.2PC	9/28/2008	6	6	6
55	08CIM182.2PC	9/28/2008	7	5	5
56	08CIM182.2PC	9/28/2008	8	4	4
57	08CIM182.2PC	9/28/2008	9	3	3
58	08CIM182.2PC	9/28/2008	10	2	2
59	08CIM182.2PC	9/28/2008	11	1	1
60	08CIM182.2PC	9/28/2008	12	1	1
61	08CIM182.2PC	9/28/2008	13	1	1
62	08CIM182.2PC	9/28/2008	14	1	1
63	08CIM182.2PC	9/28/2008	15	1	1
64	08CIM182.2PC	9/28/2008	16	1	1
65	08CIM182.2PC	9/28/2008	17	1	1
66	08CIM182.2PC	9/28/2008	18	2	2
67	08CIM182.2PC	9/28/2008	19	3	3
68	08CIM182.2PC	9/28/2008	20	4	4
69	08CIM182.2PC	9/28/2008	21	5	5
70	08CIM182.2PC	9/28/2008	22	4	4
71	08CIM182.2PC	9/28/2008	23	5	5
72	08CIM182.2PC	9/28/2008	24	6	6