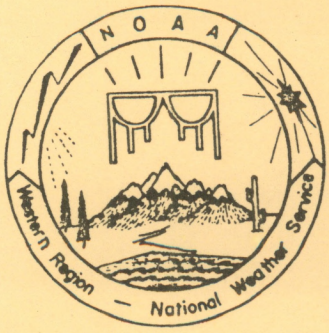
estern Region Computer Programs and Problems NWS WRCP - No. 8

RAOB PLOT AND ANALYSIS ROUTINES

John Jannuzzi

National Weather Service Western Region
Portland, Oregon
January 1980

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / National Weather Service



PREFACE

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

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

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

NOAA Western Region Computer Programs and Problems NWS WRCP

1 Standard Format for Computer Series. June 1979
2 AFOS Crop and Soil Information Report Program. Ken Mielke, July 1979
3 Decoder for Significant Level Transmission of Raobs. John Jannuzzi, August79
4 Precipitable Water Estimate. Elizabeth Morse, October 1979
5 Utah Recreational Temperature Program. Kenneth M. Labas, November 1979
6 Normal Maximum/Minimum Temperature Program for Montana. Kenneth Mielke,Dec.79
7 Plotting of Ocean Wave Energy Spectral Data. John R. Zimmerman, December,79
8 Raob Plot and Analysis Routines. John Jannuzzi, January 1980

NOAA Western Region Computer Programs and Problems NWS WRCP - No. 8

RAOB PLOT AND ANALYSIS ROUTINES

John Jannuzzi
Weather Service Forecast Office
Portland, Oregon
January 1980

UNITED STATES
DEPARTMENT OF COMMERCE
Juanita M. Kreps, Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Richard A. Frank, Administrator

National Weather
Service
Richard E. Hallgren, Director

CONTENTS

RAOB PLOT AND ANALYSIS ROUTINES

John Jannuzzi
National Weather Service Forecast Office
Portland, Oregon

I. GENERAL INFORMATION

A. Summary:

A set of three routines is described that run from decoded raob information on AFOS. The first routine is a vertical profile plot of the temperature, dewpoint and wind. The second routine is a comprehensive convective analysis; computation of a number of convective parameters and indices used to forecast potential for convective activity. The third routine evaluates the sounding for layers of strongest wind shear. All or any one of these routines can be selected for execution on a particular raob run at the discretion of the forecaster from the AFOS console.

B. Environment:

The programs are written in Data General Fortran IV. They have been designed to run on the Eclipse S-230 minicomputer of AFOS. Each routine (program) can be executed in less than 10K memory; hence, each can run in the background partition (Background) of any WSO or WSFO while AFOS is running in the foreground partition (Foreground). To do this, a shortened version of the plot was written. Where background is not limited to 10K (13K needed) a more complete version of the plot routine is available.

C. References:

The program utilizes two decoded raob files as input. Both of these are created by another program ("SGLDECODER.SV") which decodes the second transmission of the raob (for a complete description of these files and the decode program "SGLDECODER" see NOAA Western Region Computer Programs and Problems NWS WRCP No. 3).

These programs have undergone a few months testing period to allow programming errors to surface. The plotting routine has undergone extensive testing in three NWS regions. To help insure that the programs performed tasks useable to a wide variety of NWS users, routines for the convective analysis were discussed with personnel from the National Severe Storms Forecast Center and the wind shear analysis was discussed with the Weather Service Evaluation Officer at SLC WSFO.

II. APPLICATION

A. Complete Program Description

1. PLOT routine (13K version for WSFO's named "FOPLOTRAOB.FR")

This program takes the decoded sounding information, and puts it into a plot form that can be displayed on AFOS as a graphic. Five subroutines are used to create the AFOS graphic format. These subroutines are part of a graphics library created at the Scientific Services Division, Western Region NWS. These are to be described in a forthcoming WRCP.

The program begins by opening the input file containing decoded temperature and dewpoint depression information (SOUNDING.T). It first reads the station number, date, and time of the raob. Next, it reads the pressure and temperature of each reported layer and assigns that report an X-Y coordinate location for the plot. The plot uses a linear temperature scale which goes from -70°C to +50°C. This temperature scale occurs horizontally over a length of 3000 pixels on the graphics screen. The horizontal pressure scale of pressure to the .2857 power ($p^{.2857}$), goes from the surface to 100 mb. The range from 1000 mb to 100 mb occurs over a length of 2500 pixels. The file "SOUNDING.T" is then closed and reopened. A similar sequence of events to the above is repeated to assign the dewpoint/pressure coordinates for the plot. "SOUNDING.T" is then closed.

The main program calls a subroutine named "KINDEX". This subroutine computes the K Index for the raob. In order to do this, using only the second transmission of the raob, a linear interpolation is performed between reported levels to derive temperature and dewpoint values at 500, 700 and 850 mb. The linear interpolation is performed on the assigned plot values, not on the actual pressure and temperatures values themselves, as the pressure axis is not linear. Again, the information is read in from the file "SOUNDING.T". The subroutine returns the K index under the variable name "K". If one of the parameters necessary for the K index computation is not available, a K of -1000 is returned to the program. The program then puts the K index into graphic output format.

The next section of the program puts the station number, date, and time information in graphic format. Any integer number must be handled one digit at a time. Each integer digit must be converted to an alphnumeric representation (A-format) for display. This is done by multiplying the digit by 256 and adding to that quantity the integer 12288.

The last section of the program puts the wind information, from the file "SOUNDING.W", into a plot form. The main problem was to convert the height levels reported in the wind portion of the raob to pressure levels on the sounding plot. To do this, the heights of the

pressure levels in the U.S. Standard Atmosphere were used. The values for every 100 mb decrement of pressure starting at 1000 mb were read off the Air Force Skew-T/Log P diagram. These values are stored in hundreds of feet in the array "IINT" in the program. The program reads the height level, wind direction and speed. It assigns the pressure value by interpolating between the levels in the array "IINT". The direction is checked to see if it is a multiple of five; if not, 100 is added to the wind speed. The wind arrow is drawn so that the head is along the pressure axis and the tail is at the direction from which the wind is blowing (standard meteorological practice). The exact wind speed is written beside the tail of the arrow.

2. (10K version for WSO's named "PLOTRAOB1.FR" and "PLOTRAOB2.FR")

Except for a few changes, this is essentially the same as "FOPLOTRAOB.FR". In order to fit this in 10K, the program was split into two pieces. The first piece automatically calls in (SWAPs to) the second piece. To do this, some intermediate values must be temporarily stored on disk from the first piece and read into the second piece. These are put into a file name "DPØ:SCRATCH.NW". This file is automatically created when needed and deleted when no longer needed. The variables "NW" and "SFCPRS" are saved into this file.

Another change is that the steps to compute the K index are contained within the first piece itself, rather than in a separate subroutine. The final difference is a check to see how large "NW" (NW is a counter in the program and is related to the number of vectors in the graphic) gets to be. The variable "NS" and "NS2" are dimensioned sufficiently large to handle long raobs in "FOPLOTRAOB.FR". Due to size constraints this could not be done in the 10K program version. Hence, whenever "NW" is greater than 472 the program terminates. The result of this is that on long raob transmissions, some of the highest levels of winds will be left off the plot.

3. Convective Analysis ("INDEXES.FR")

This program computes the K Index, Showalter Index, Lifted Index Lifted Condensation Level and its temperature, Level of Free Convection, Convective Condensation Level and its temperature, Convective temperature, and energy in the positive area of the sounding.

The program begins by reading from the file "SOUNDING.T" all the pressure, temperature, and dewpoint depression values. These are stored into 3 arrays: "PRESS", "TEMP", and "DEPT" respectively. The program then interpolates for temperature and dewpoint values at 850, 700, and 500 mb via a subroutine named "INTERPOLATE". This subroutine performs linear interpolations in the same manner as described above for computation of the K index. The interpolated values are passed to the main program as "TMP" amd "DEW". The last variable passed in

the statement call is a flag to indicate whether or not the dewpoint interpolation is necessary. If this is "1", only a temperature interpolation is performed. If the dewpoint at a level necessary for the interpolation is missing a "DEW" of -99 is returned. Likewise, if no temperature is available "TMP" is returned as -99.

The first quantity calculated is the Showalter Index. The computation begins by taking the 850 mb temp and dewpoint as a parcel and raising it to saturation (the LCL). The values for this are stored as the variables TLCL (temperature of LCL) and PLCL (pressure at LCL). Next the parcel is lifted moist adiabatically until the parcel reaches 500 mb. This is done in a subroutine named "LIFT". "LIFT" takes as input an initial pressure and temperature of a saturated parcel and its final pressure after lifting, and returns to the main program the final parcel temperature. If the parcel from 850 mb does not reach saturation before reaching 500 mb the call to "LIFT" is bypassed. The Showalter Index is computed as "SHWLT", and is rounded off to the nearest integer in a short subroutine named "MKRND". The final Showalter Index is the variable "ISHWLT".

The next parameter derived is the Lifted Index. It is begun by computing a mean mixing ratio in the lowest 50 mb of the sounding in a subroutine named "MMXRT". This is converted to an average dewpoint at the surface and returned to the main program as "TD". The maximum temperature for the day is computed by adding 2°C to the 700 mb temperature, and bringing a parcel at that temperature and pressure dry adiabatically to the surface. A flag, "II", is changed to "1", indicating that the Lifted Index rather than the Showalter is to be computed, and the program loops back toward the beginning. This surface parcel is now raised to saturation and then to 500 mb for comparison with the actual raob temperature at 500 mb. The Lifted Index, after being rounded off, is stored as "LFTIND".

The K Index is easily computed in the next few steps, as the necessary values at 500, 700 and 850 mb are already available. This is stored as "K".

The next parameter calculated is the Convective Condensation Level. This is calculated by taking the mean dewpoint, "TD", used above for the Lifted Index, converting it to a mixing ratio and finding where the mixing ratio line crosses the temperature sounding. The mixing ratio is calculated as the variable "COMXRT". To find it, the vapor pressure is needed. It is calculated as the variable "EE". The parcel (mixing ratio value) is ascended 10 mb at a time, the temperature is calculated, and it is compared to the sounding temperature. The sounding temperature is found by a linear interpolation using the subroutine INTRPOLATE as before. Whenever the parcel temperature is found to be warmer than or equal to the sounding temperature ("COMPR" ≤ 0), the exact crossing is computed by a similar triangle approach (interpolation). This is then defined as the CCL. The integer value is found through the subroutine "MKRND" and stored as "ICCL" (pressure value). This pressure value

is entered into "INTRPOLATE" and the temperature of the CCL is returned to the main program. This is stored as "TCCL".

The trigger temperature (temperature of the surface at which convection will begin) is calculated by simply bringing the parcel at the CCL dry adiabatically to the surface. The dry adiabatic formula is stored as "THETA" and the necessary surface temperature as "TCONV".

The Level of Free Convection and the Lifted Condensation Level calculations are performed in a separate subroutine named "LFC". The subroutine begins by refiguring the mean low level dewpoint with a call to "MMXRT". This is used in the equations for "TLCL" and "PLCL" to find the temperature and pressure, respectively, of the Lifted Condensation Level. These are rounded off to integers at the end of the subroutine and stored as "LCLTMP" and "LCLPRS". These LCL values are then used to find the LFC.

The LFC is found by raising the parcel at the LCL moist adiabatically until it crosses the temperature sounding. This is done similarly to the method described above for the CCL computation. The parcel is raised 10 mb at a time through the subroutine "LIFT". Each time, the parcel temperature is compared to the sounding temperature as "COMPR". When "COMPR" ≤ 0 the lift is terminated, the exact crossing interpolated for, and the LFC is stored as "ALFC". The rounded off integer value is stored as "ILFC". If the parcel rise reached 500 mb before the parcel is warmer than the temperature sounding "ILFC" is assigned a value of -99. The quantities "ILFC", "LCLTMP" and "LCLPRS" are returned to the main program.

The final computation is the energy in the positive area of the sounding. The parcel is raised from the CCL until it becomes cooler than the temperature sounding. This area is found by summing up small areas between the sounding temperature, parcel temperature, and 10 mb pressure levels. "TAREA" is the equation for this area (in units of joules/gram).

The end of the program is the output of the above derived quantities. The output is placed into a file named "INDEXOUT". If such a filename already exists, it is deleted and a new one is created.

4. Windshear Analysis ("WINDSHEAR.FR")

This program computes the vector wind shear between all reported wind levels of the second transmission of the raob (provided the wind speeds are 10 knots or greater). It begins by reading in all the information from the files "SOUNDING.T" and "SOUNDING.W". The program then enters a large loop to compute the windshears. The windshear is computed in the subroutine "COMSHR". It separates each wind direction and speed into its N-S and E-W components and computes the absolute vector wind shear. This wind shear is reduced

to its value in knots per 1000 feet and returned to the main program as "WNDSHR". The integer round off value is found in "MKRND" and stored as "IWNDSH". The Richardson number for that layer is also computed.

To compute the Richardson number, two interpolations are performed. The first converts the height levels of the wind reports to pressure values using the same interpolation for the Standard Atmosphere value as discussed in the raob plot program (section II-A-1). This is performed in a subroutine name "PRSINT". The second interpolation is for the temperature at the pressure level. This, as discussed in previous sections, is done via the subroutne "INTRPOLATE". The temperatures are converted to potential temperatures ("THETA1" and THETA2") and a mean potential temperature is computed ("THETAM"). The layer thickness, "DZ", is simply the higher level minus the lower level. "DTHETA" is "THETA2" minus "THETA1". These quantities are used to compute the Richardson Number ("RNUM").

The rest of this DO loop is a sort test performed on the windshear, "IWDNSH". The windshears are ranked in decreasing order in the array "IWSHR". Only the highest five windshears are saved. If the one just calculated is greater than any of the previously saved windshears, the new one replaced the old one in "IWSHR" and moves the rest down one position in the ranking. If they are the same, the newer one is placed below the old one, so that the highest shears at the lowest levels are ranked first at all times. This is done in a subroutine named "BUMP". When this is completed, the cycle begins again for the next two levels of wind.

When all the wind shears have been computed and ranked, the five highest windshears and their Richardson numbers are written out to a file named "WINDSHEAR". If a file with that name already exists, it is deleted and a new one created.

B. Machine Requirements

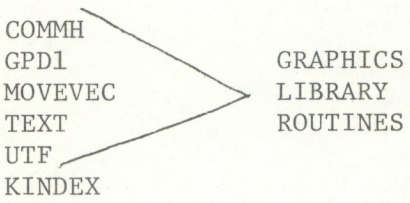
The "FOPLOTRAOB" routine requires 13K memory to run. The WSO version of this, "PLOTRAOB1" and "PLOTRAOB2", can run in 10K. "KINDEX" and "WINDSHEAR" can each run in 10K. The following table lists the save files (.SV extensions), their disk storage requirement (size), and average execution times from the floppy and rigid disks.

		EXECUTION TIME	
SAVE FILE	SIZE (bytes)	FLOPPY DISK	RIGID DISK
FOPLOTRAOB.SV	18,944	3/4 - 1 minute	15 - 25 seconds
PLOTRAOB1.SV	16,384		
PLOTRAOB2.SV	15,360	1 - 1 1/4 minutes	35 - 55 seconds
INDEXES.SV	17,920	1 - 3 minutes	3/4 - 2 1/2 minutes
WINDSHEAR.SV	16,384	3/4 - 1 minute	10 - 15 seconds

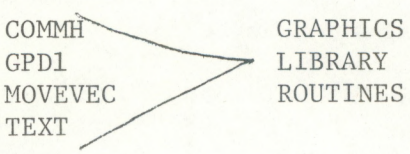
C. Structure of Software

There are a number of subroutines that are used for each of the routines. These were briefly discussed in the descriptions of the main programs above. The following lists show the main program title and the names of all the subroutines that are required by it.

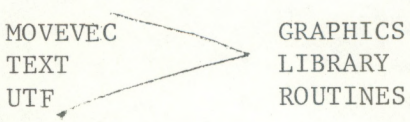
1. FOPLOTRAOB



2. PLOTRAOB1



3. PLOTRAOB2



4. INDEXES

INTRPOLATE
LIFT
MKRND
MMXRT
LFC

5. WINDSHEAR

COMSHR
MKRND
PRSINT
INTRPOLATE
BUMP

D. Data Base

All these routines use the files "SOUNDING.T" and "SOUNDING.W" as input files. No directory specifiers are used, so these files must be in the same directory as the executable save files (.sv files).

The output files are all created in the programs themselves. If files with those names exist already, they are deleted and new ones are created. The output file for "FOPLOTRAOB" and for the two program pair of "PLOTRAOB1" and "PLOTRAOB2" is "RAOBPLOT". The output file for "INDEXES" is "INDEXOUT" and for "WINDSHEAR" is "WINDSHEAR".

In order that the plotted raob be of useable form, separate map background files (grids) have been created. These could have been part of the plot program itself, but then the plot and backgrounds could not have been called on separate overlays on a graphics screen. The grid containing pressure, temperature, and dry adiabat contours on it is in a file named "RAOBGRID". This same grid, with the addition of mixing ratio contours, is in "RAOBGRID.MR". The moist adiabats are in a file names "MOIST".

III. PROCEDURES

A. Initiation of Program/B. Input Requirements (see also Appendix)

All of the routines require that the input files be in the same directory as the routines themselves. Also, the routines must be initiated from the directory where they reside.

Since all the routines use "SOUNDING.T" and "SOUNDING.W" files as input, the raob decoder program, "SGLDECODER", must be run before these routines are initiated (see WRCP No. 3 for instructions on initiating "SFLDECODER").

C. Output

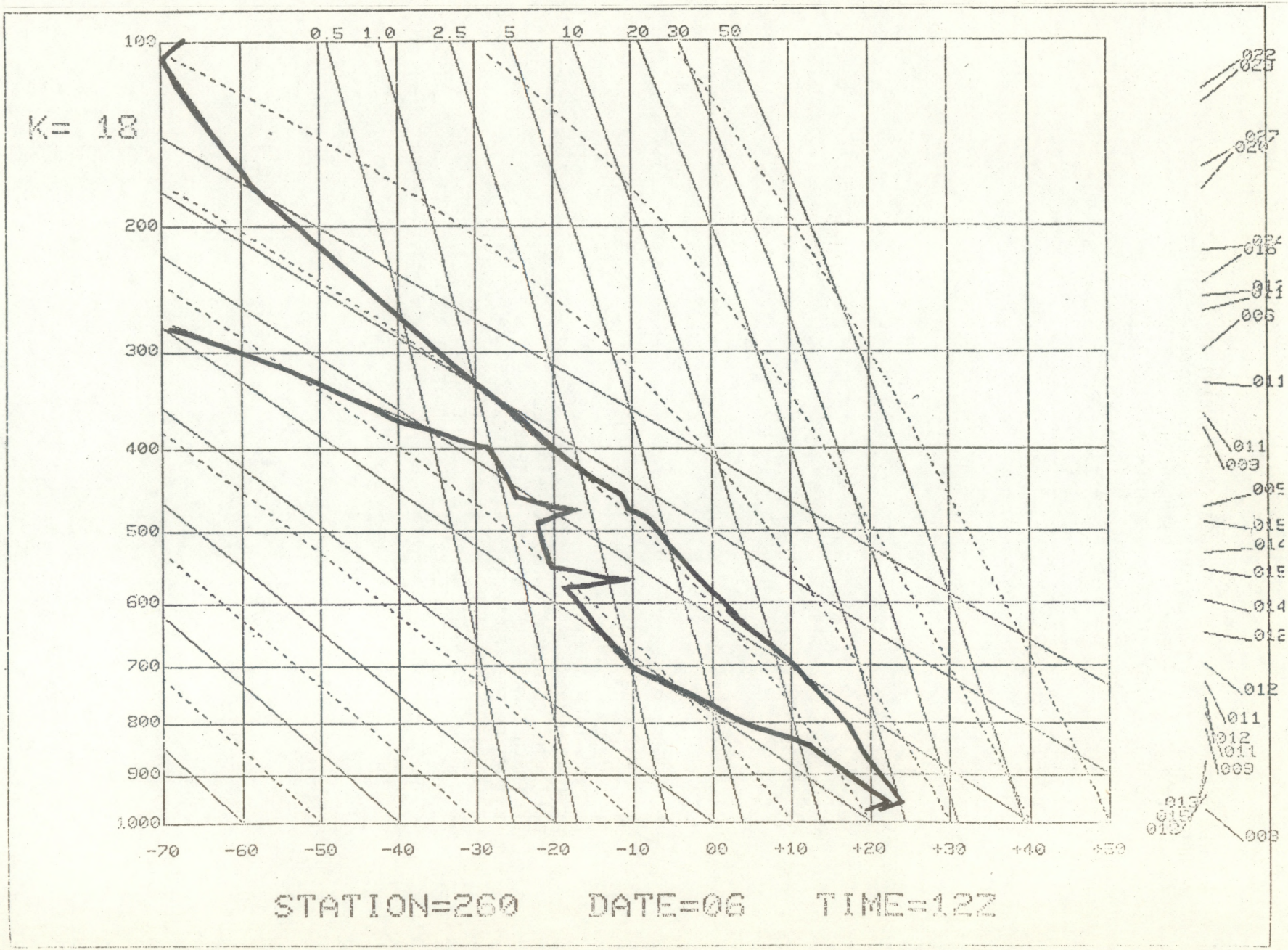
The output files "INDEXOUT" and "WINDSHEAR" are random RDOS files that can be displayed on an AFOS ADM (i.e., DSP:DP1:INDEXOUT). The output file "RAOBPLOT" is a graphic file that can be displayed on an AFOS GDM (DSP:DP1:RAOBPLOT). Any of these output files can be stored into the AFOS data base (i.e., STORE:DP1:RAOBPLOT SLCRABSLC) and called up directly from the data base. This allows the routines to be rerun on a different sounding and have the output from the previous run still available for examination.

The following is an example of a raob second transmission and the resultant files "SOUNDING.W" and "SOUNDING.T" created by running "SGLDECODER". Also displayed are examples of "RAOBPLOT", "INDEXOUT" and "WINDSHEAR" from that sounding.

```
R
TYPE RAOB.BB
WRHTSTJAJ           EWOUS00 KWRH 111829
 DRT 72260 TTBB  5612/ 72260 00974 21015 11960 24014 22850 19856
33888 17264 44700 10271 55576 00968 66565 01958 77551 02957
88487 08164 99472 10556 11455 11563 22400 20558 33373 24163
44275 38780 55176 585// 66106 699// 77100 669//=
PPBB  56120 72260 90023 13003 21518 20015 90467 19013 16509
16011 9069/ 17012 15511 91024 13012 10012 10514 91679 10015
08514 10015 92056 07509 15509 14511 928// 09511 93034 05006
07511 08513 9357/ 05516 08524 94249 04020 06027 05023 9504/
05522 09015=
```

```
TYPE SOUNDING.T
260  6 12
 974  21.0  1.5
 960  24.0  1.4
 850  19.0  6.0
 800  17.2 14.0
 700  10.2 21.0
 576  -0.9 18.0
 565  -1.9  8.0
 551  -2.9 17.0
 487  -8.1 14.0
 472 -10.5  6.0
 455 -11.5 13.0
 400 -20.5  8.0
 373 -24.1 13.0
 275 -38.7 30.0
 176 -58.5 -1.0
 106 -69.9 -1.0
 100 -66.9 -1.0
R
```

```
TYPE SOUNDING.W
260  6 12
 0 130 3
 2 21518
 3 20015
 4 19013
 6 165 9
 7 16011
 8 17012
 9 15511
10 13012
12 10012
14 10514
16 10015
17  8514
19 10015
20  75 9
25 155 9
26 14511
28  9511
30  50 6
33  7511
34  8513
35  5516
37  8524
42  4020
44  6027
49  5023
50  5522
54  9015
R
```


K= 18
0.5 1.0 2.5 5 10 20 30 50
100
200
300
400
500
600
700
800
900
1000
-70 -60 -50 -40 -30 -20 -10 00 +10 +20 +30 +40 +50
STATION=260 DATE=06 TIME=12Z

THE 5 LAYERS WITH THE STRONGEST WIND SHEAR! IN DECREASING ORDER, ALONG WITH THE LAYER'S RICHARDSON NUMBER ARE:

	LAYER 1000'S FT	SHEAR KNOTS/1000FT	RICHARDSON NUMBER
1	34 - 35	8	0.71
2	35 - 37	6	1.30
3	2 - 3	5	2.32
4	42 - 44	5	2.71
5	26 - 28	5	2.03

PAGE 01

STATION = 260 DATE = 6 TIME = 12
THE K INDEX = 18
THE SHOWALTER INDEX = 0
THE LIFTED INDEX = -7
THE LIFTED CONDENSATION LEVEL IS AT 928MB
AND HAS A TEMPERATURE OF 17 DEGREES C
NO LEVEL OF FREE CONVECTION EXISTS BELOW 500MB
THE CONVECTIVE CONDENSATION LEVEL IS AT 745MB
AND HAS A TEMPERATURE OF 14 DEGREES C
CONVECTION WILL BEGIN WHEN THE SURFACE TEMPERATURE REACHES 36 DEGREES C
THE ENERGY IN THE POSITIVE AREA OF THE RAOB IS 2.161JOULES/GRAM

PAGE 0

D. Cautions and Restrictions

These routines will work on any raob. The trigger temperature value, however, in "INDEXOUT" is designed for 12Z raobs only. This is because it uses a uniform warming factor to approximate the maximum temperature of the day. If run at 00Z (near max heating time) an artificially high maximum temperature value will be created. Also, the heating factor was designed for summer use and assumes much solar radiation will reach the surface. It will not be very good in a cloudy condition or in winter.

These routines use only input from the second transmission of a raob. Hence, values like the K index may be slightly different than those computed directly from the first transmission of the raob. Under most instances, the interpolated values of temperatures at mandatory levels will be very close to the actual values, especially in moist soundings when the numbers are most needed.

E. Complete Program Listing

The following is a listing of the main programs and subroutines used, except for the graphics library subroutines.

```
TYPE FOPLOTRAOB.FR
C*****RAOB PLOTTING PROGRAM FOR EXECUTION IN 13K
      PARAMETER NS=768,NS2=1536
      DIMENSION IGA(NS),IT(25),IVEC(75),IINT(10),IB(NS2)
      IT(1)="NM"
      IT(2)="CG"
      IT(3)="PH"
      IT(4)="WR"
      IT(5)="1 "
      CALL COMMH(IGA,IT,NW,NS)
      CALL GPD1(IGA,NW,NS)
      CALL OPEN(6,"SOUNDING.T",2,IER)
      READ(6,12) ISTATN,IDATE,ITIME
   12 FORMAT(I3,1X,I2,1X,I2)
      N=1
C*****READS THE PRESSURE LEVEL AND IT'S TEMPERATURE, THEN CONVERTS
C*****THIS TO A POINT ON THE PLOT.
   50 READ(6,25,END=75) IPRESS,TEMP
   25 FORMAT(I4,1X,F5.1)
      NODD=(N*2)-1
      NEVEN=(N*2)
      IF(IPRESS .LT. 100) IPRESS=IPRESS+1000
      IF(N .EQ. 1) SFCPRS=IPRESS
      PRESS=FLOAT(IPRESS)
      IVEC(NEVEN)=IFIX(((7.196-PRESS**.2857)*720.71)+400.0))
      IVEC(NODD)=IFIX((TEMP+70.0)*25.0)+500.0)
      N=N+1
      GO TO 50
   75 N=(N*2)-1
      IVEC(N)=999
      CALL CLOSE(6,IER)
C*****DRAWS THE VECTORS TO ALL THE POINTS ON THE TEMPERATURE PROFILE.
      CALL MOVEVEC(IGA,IVEC,NW,NS)
      CALL OPEN(6,"SOUNDING.T",2,IER)
      N=1
      READ(6,12) ISTATN,IDATE,ITIME
C*****READS THE PRESSURE LEVEL, IT'S TEMPERATURE AND DEWPOINT DEPRESSION,
C*****THEN CONVERTS THIS TO A DEWPOINT AND PUTS THE POINT ON THE PLOT.
  150 READ(6,125,END=175) IPRESS,TEMP,DEPR
  125 FORMAT(I4,1X,F5.1,1X,F4.1)
      NODD=(N*2)-1
      NEVEN=(N*2)
      IF(DEPR .EQ. -1.0) GO TO 150
      PRESS=FLOAT(IPRESS)
      IVEC(NEVEN)=IFIX(((7.196-PRESS**.2857)*720.71)+400.0))
      IVEC(NODD)=IFIX((TEMP-DEPR+70.0)*25.0)+500.0)
      N=N+1
      GO TO 150
  175 N=(N*2)-1
      IVEC(N)=999
      CALL CLOSE(6,IER)
C*****DRAWS THE VECTORS TO ALL THE POINTS ON THE DEWPOINT PROFILE.
      CALL MOVEVEC(IGA,IVEC,NW,NS)
      NSIZE=3
      IL=80
      JL=2600
C*****COMPUTES THE K INDEX.  A K OF -1000 MEANS IT WAS NOT ABLE TO
C*****BE CALCULATED.
      CALL KINDEX(K)
```

```
C*****PUTS THE K INDEX INTO THE PLOT.
      IF(K .EQ. -1000) GO TO 656
      IT(1)="K="
      IT(2)=" "
      IF(K .LT. 0) IT(2)="-"
      K=IABS(K)
      IT(3)=((K/10)*256)+12288
      IT(4)=((K-((K/10)*10))*256)+12288
      IT(5)=0
      GO TO 686
  656 IT(1)="K="
      IT(2)=" "
      IT(3)="NA"
      IT(4)=0
  686 CALL TEXT(NSIZE,IGA,IL,JL,IT,NW,NS)
C*****PUTS THE STATION NUMBER, DATE AND TIME INTO THE PLOT.
      IL=850
      JL=100
      IT(1)="ST"
      IT(2)="AT"
      IT(3)="IO"
      IT(4)="N="
      IHUND=ISTATN/100
      IT(5)=(IHUND*256)+12288
      IHUND=100*IHUND
      ITEN=(ISTATN-IHUND)/10
      IT(6)=(ITEN*256)+12288
      ITEN=10*ITEN
      IT(7)=((ISTATN-IHUND-ITEN)*256)+12288
      IT(8)="  "
      IT(9)=" "
      IT(10)="DA"
      IT(11)="TE"
      IT(12)="="
      ITEN=IDATE/10
      IT(13)=(ITEN*256)+12288
      ITEN=10*ITEN
      IT(14)=((IDATE-ITEN)*256)+12288
      IT(15)="  "
      IT(16)=" "
      ITEN=ITIME/10
      IT(17)="TI"
      IT(18)="ME"
      IT(19)="="
      IT(20)=(ITEN*256)+12288
      ITEN=10*ITEN
      IT(21)=((ITIME-ITEN)*256)+12288
      IT(22)="Z"
      IT(23)=0
X     TYPE NSIZE,IL,JL,NW
      CALL TEXT(NSIZE,IGA,IL,JL,IT,NW,NS)
C*****AN ARRAY ASSIGNING U.S. STANDARD ATMOSPHERE HEIGHTS TO THE
C*****PRESSURE LEVELS AT 100MB DECREMENTS.
      IINT(1)=04
      IINT(2)=32
      IINT(3)=64
      IINT(4)=99
      IINT(5)=138
      IINT(6)=184
      IINT(7)=236
      IINT(8)=300
      IINT(9)=386
      IINT(10)=532
```

```
      CALL OPEN(8,"SOUNDING.W",2,IER)
C*****READS IN THE HEIGHT AND IT'S WIND SPEED AND DIRECTION.
      READ(8,12) ISTATN,IDATE,ITIME
  400 READ(8,401,END=300)ILVL,IDIR,ISPEED
  401 FORMAT(I2,1X,I3,I2)
C*****ASSIGNS THE HEIGHT LEVEL A PRESSURE VALUE, THEN A "Y"
C*****COORDINATE LOCATION ON THE PLOT.
      ILVL=ILVL*10
      IF(ILVL .LT. IINT(1)) PRESS=1010.0
      IF(ILVL .EQ. 0) PRESS=SFCPRS
      IF(ILVL .LT. IINT(1)) GO TO 450
      DO 425 IJ=2,10
      IJM1=IJ-1
      A= FLOAT(ILVL) - FLOAT(IINT(IJM1))
      B=FLOAT(IINT(IJ)) - FLOAT(IINT(IJM1))
      A=(A/B)*100.0
      B=1100. - ((FLOAT(IJM1))*100.0)
      IF(ILVL .LT. IINT(IJ)) PRESS= B - A
      IF(ILVL .LT. IINT(IJ)) GO TO 450
  425 CONTINUE
  450 NP1=IFIX((((7.196-PRESS**.2857)*720.71)+400.0))
      IVEC(1)=3800
      IVEC(2)=NP1
C*****ASSIGNS THE POINT FOR THE TAIL OF THE WIND VECTOR (ARROW).
      IDIR1=((IDIR/10)*10)
      IDIR5 = IDIR + 5
      IDIR51 = ((IDIR5/10)*10)
      IF(IDIR .NE. IDIR1 .AND. IDIR5 .NE. IDIR51) ISPEED=ISPEED+100
      ANGLE=((360.-(FLOAT(IDIR)))+90.)/57.29577
      AJ=150.*SIN(ANGLE)
      JL=(IFIX(AJ))+NP1
      AI=150.*COS(ANGLE)
      IL=(IFIX(AI))+3800
      IVEC(3)=IL
      IVEC(4)=JL
      IVEC(5)=999
C*****DRAWS THE WIND VECTOR ON THE PLOT.
      CALL MOVEVEC(IGA,IVEC,NW,NS)
C*****ASSIGNS THE WIND SPEED AND IT'S LOCATION ON THE PLOT.
      NSIZE=1
      IL=IL-100
      IF(IDIR .LT. 180) IL=IL+110
      IHUND=ISPEED/100
      IT(1)=(IHUND*256)+12288
      IHUND=IHUND*100
      ITEN=(ISPEED-IHUND)/10
      IT(2)=(ITEN*256)+12288
      ITEN=ITEN*10
      IONE=(ISPEED-IHUND-ITEN)
      IT(3)=(IONE*256)+12288
      IT(4)=0
X     TYPE NSIZE,IL,JL,NW
C*****PUTS THE WIND SPEED INTO THE PLOT.
      CALL TEXT(NSIZE,IGA,IL,JL,IT,NW,NS)
      GO TO 400
  300 CALL CLOSE(8,IER)
X     TYPE NW
C*****PUTS THE PLOT INTO FINAL GRAPHIC FORM FOR DISPLAY.
      CALL UTF("RAOBPLOT",IGA,IB,NW,NS,NS2)
      STOP
      END
```

```
TYPE KINDEX.FR
      SUBROUTINE KINDEX(K)
C*****TAKES THE SOUNDING.T FILE, INTERPOLATES TEMPERATURES AND
C*****DEWPOINTS FROM REPORTED LEVELS TO 500, 700 AND 850MB, AND
C*****FINALLY COMPUTES THE K INDEX.
      DIMENSION APRESS(3),ATEMP(3),DEW(3),Y(5),X(5)
      APRESS(1)=500.0
      APRESS(2)=700.0
      APRESS(3)=850.0
      CALL OPEN(6,"SOUNDING.T",2,IER)
      I=3
C*****READS THE DATA FROM SOUNDING.T.
      READ(6,12) ISTATN, IDATE, ITIME
   12 FORMAT(I3,1X,I2,1X,I2)
      READ(6,625)PRESS,TEMP,DEPR
  625 FORMAT(F4.0,1X,F5.1,1X,F4.1)
  630 FORMAT(1X,"MOISTURE DATA NOT AVAIBLABLE FOR THIS CALCULATION")
C*****CHECKS TO SEE IF THE MOISTURE INFORMATION WAS REPORTED (A -1
C*****INDICATES MISSING DATA).
      IF(DEPR.EQ. -1.0) WRITE(10,630)
      IF(DEPR .EQ. -1.0) GO TO 655
C*****BEGINS A SCAN OF THE DATA FOR THE INTERPOLATION.
      IF(PRESS .LT. APRESS(I)) WRITE(10,631)
  631 FORMAT(1X,"SURFACE LEVEL ABOVE 850MB")
      IF(PRESS .LT. APRESS(I)) GO TO 655
      IF(PRESS .EQ. APRESS(I)) ATEMP(I)=TEMP
      IF(PRESS .EQ. APRESS(I)) DEW(I)=TEMP-DEPR
      IF(PRESS .EQ. APRESS(I)) GO TO 628
  620 PRESS1=PRESS
      TEMP1=TEMP
      DEPR1=DEPR
      READ(6,625,END=655)  PRESS,TEMP,DEPR
  622 IF(PRESS .EQ. APRESS(I)) ATEMP(I) = TEMP
      IF(PRESS .EQ. APRESS(I)) DEW(I) = TEMP-DEPR
      IF(PRESS .EQ. APRESS(I)) GO TO 628
      IF(APRESS(I) .GT. PRESS .AND. APRESS(I) .LT. PRESS1) GO TO 610
      GO TO 620
C*****PERFORMS THE TEMPERATURE INTERPOLATION.
  610 Y(4)=((7.196-(PRESS1**.2857))*720.71)
      Y(5)=((7.196-(PRESS**.2857))*720.71)
      X(4)=(((TEMP1+70.0)*25.0)+500.0)
      X(5)=(((TEMP+70.0)*25.0)+500.0)
X     TYPE "Y(4)=",Y(4),"Y(5)=",Y(5),"X(4)=",X(4),"X(5)=",X(5)
X     TYPE "SLOPE=",SLOPE
      IF((X(5)-X(4)) .EQ. 0.0) X(5) = X(5) + .1
      SLOPE=(Y(5)-Y(4))/(X(5)-X(4))
      Y(I)=((7.196-(APRESS(I)**.2857))*720.71)
      X(I)=(((Y(I)-Y(5)))/SLOPE)+X(5)
X     TYPE "Y(I)=",Y(I),"X(I)=",X(I)
      ATEMP(I)=((X(I)-500.0)/25.0)-70.0
C*****PERFORMS THE DEWPOINT INTERPOLATION.
      X(4)=(((TEMP1-DEPR1+70.0)*25.0)+500.0)
      X(5)=(((TEMP-DEPR+70.0)*25.0)+500.0)
      IF((X(5) - X(4)) .EQ. 0.0) X(5) = X(5) + .1
      SLOPE=(Y(5)-Y(4))/(X(5)-X(4))
X     TYPE "X(4)=",X(4),"X(5)=",X(5),"Y(4)=",Y(4),"Y(5)=",Y(5)
X     TYPE "SLOPE=",SLOPE
      X(I)=((Y(I)-Y(5))/SLOPE)+X(5)
X     TYPE "X(I)=",X(I)
      DEW(I)=((X(I)-500.0)/25.0)-70.0
X     TYPE "I= ",I," ATEMP(I)= ",ATEMP(I)," DEW(I)= ",DEW(I)
  628 I=I-1
      IF(I .EQ. 0) GO TO 650
      GO TO 622
C*****CALCULATES THE K INDEX
  650 AK=-ATEMP(1)+ATEMP(3)+DEW(3)-(ATEMP(2)-DEW(2))+.5
      K=IFIX(AK)
      GO TO 685
  655 K=-1000
  685 CALL CLOSE(6,IER)
      RETURN
      END
```

```
TYPE PLOTRAOB1.FR
C*****FIRST PART OF THE RAOB PLOTTING PROGRAM FOR EXECUTION IN 10K.
      PARAMETER NS=512,NS2=1024
      DIMENSION IGA(NS),IT(50),IVEC(100),APRESS(3),ATEMP(3),DEW(3),
     1Y(5),X(5)
      APRESS(1)=500.0
      APRESS(2)=700.0
      APRESS(3)=850.0
      IT(1)="NM"
      IT(2)="CG"
      IT(3)="PH"
      IT(4)="WR"
      IT(5)="1 "
      CALL COMMH(IGA,IT,NW,NS)
      CALL GPD1(IGA,NW,NS)
      CALL OPEN(6,"SOUNDING.T",2,IER)
      READ(6,12) ISTATN,IDATE,ITIME
   12 FORMAT(I3,1X,I2,1X,I2)
      N=1
C*****READS THE PRESSURE LEVEL AND THE TEMPERATURE, THEN CONVERTS THIS
C*****TO A POINT ON THE PLOT.
   50 READ(6,25,END=75) IPRESS,TEMP
   25 FORMAT(I4,1X,F5.1)
      NODD=(N*2)-1
      NEVEN=(N*2)
      IF(IPRESS .LT. 100) IPRESS=IPRESS+1000
      IF(N .EQ. 1) SFCPRS=IPRESS
      PRESS=FLOAT(IPRESS)
      IVEC(NEVEN)=IFIX(((7.196-PRESS**.2857)*720.71)+400.0))
      IVEC(NODD)=IFIX((TEMP+70.0)*25.0)+500.0)
      N=N+1
      GO TO 50
   75 N=(N*2)-1
      IVEC(N)=999
      CALL CLOSE(6,IER)
C*****DRAWS THE VECTORS TO ALL THE POINTS ON THE TEMPERATURE PROFILE.
      CALL MOVEVEC(IGA,IVEC,NW,NS)
      CALL OPEN(6,"SOUNDING.T",2,IER)
      N=1
      READ(6,12) ISTATN,IDATE,ITIME
C*****READS THE PRESSURE LEVEL, IT'S TEMPERATURE AND DEWPOINT
C*****DEPRESSION, THEN CONVERTS THIS TO A DEWPOINT AND PUTS THE POINT
C*****ON THE PLOT.
  150 READ(6,125,END=175) IPRESS,TEMP,DEPR
  125 FORMAT(I4,1X,F5.1,1X,F4.1)
      NODD=(N*2)-1
      NEVEN=(N*2)
      IF(DEPR .EQ. -1.0) GO TO 150
      PRESS=FLOAT(IPRESS)
      IVEC(NEVEN)=IFIX(((7.196-PRESS**.2857)*720.71)+400.0))
      IVEC(NODD)=IFIX((TEMP-DEPR+70.0)*25.0)+500.0)
      N=N+1
      GO TO 150
  175 N=(N*2)-1
      IVEC(N)=999
      CALL CLOSE(6,IER)
C*****DRAWS THE VECTORS TO ALL THE POINTS ON THE DEWPOINT PROFILE.
      CALL MOVEVEC(IGA,IVEC,NW,NS)
      CALL OPEN(6,"SOUNDING.T",2,IER)
      NSIZE=3
      IL=80
      JL=2600
```

```
C*****COMPUTES THE K INDEX.  A K OF -1000 MEANS IT WAS NOT ABLE TO BE
C*****CALCULATED.  AN INTERPOLATION OF TEMPERATURE AND DEWPOINTS FROM
C*****REPORTED LEVELS IN SOUNDING.T IS PERFORMED FOR THE LEVELS OF 500,
C*****700 AND 850MB.
      I=3
C*****READS THE DATA FROM SOUNDING.T.
      READ(6,12) ISTATN, IDATE, ITIME
      READ(6,625)PRESS,TEMP,DEPR
  625 FORMAT(F4.0,1X,F5.1,1X,F4.1)
  630 FORMAT(1X,"MOISTURE DATA NOT AVAIBLABLE FOR THIS CALCULATION")
C*****CHECKS TO SEE IF THE MOISTURE INFORMATION WAS REPORTED ( A -1
C*****MEANS MISSING DATA).
      IF(DEPR.EQ. -1.0) WRITE(10,630)
      IF(DEPR .EQ. -1.0) GO TO 655
C*****BEGINS A SCAN OF THE DATA FOR THE INTERPOLATION.
      IF(PRESS .LT. APRESS(I)) WRITE(10,631)
  631 FORMAT(1X,"SURFACE LEVEL ABOVE 850MB")
      IF(PRESS .LT. APRESS(I)) GO TO 655
      IF(PRESS .EQ. APRESS(I)) ATEMP(I)=TEMP
      IF(PRESS .EQ. APRESS(I)) DEW(I)=TEMP-DEPR
      IF(PRESS .EQ. APRESS(I)) GO TO 628
  620 PRESS1=PRESS
      TEMP1=TEMP
      DEPR1=DEPR
      READ(6,625,END=655)  PRESS,TEMP,DEPR
  622 IF(PRESS .EQ. APRESS(I)) ATEMP(I) = TEMP
      IF(PRESS .EQ. APRESS(I)) DEW(I) = TEMP-DEPR
      IF(PRESS .EQ. APRESS(I)) GO TO 628
      IF(APRESS(I) .GT. PRESS .AND. APRESS(I) .LT. PRESS1) GO TO 610
      GO TO 620
C*****PERFORMS THE TEMPERATURE INTERPOLATION.
  610 Y(4)=((7.196-(PRESS1**.2857))*720.71)
      Y(5)=((7.196-(PRESS**.2857))*720.71)
      X(4)=(((TEMP1+70.0)*25.0)+500.0)
      X(5)=(((TEMP+70.0)*25.0)+500.0)
X     TYPE "Y(4)=",Y(4),"Y(5)=",Y(5),"X(4)=",X(4),"X(5)=",X(5)
X     TYPE "SLOPE=",SLOPE
      IF((X(5)-X(4)) .EQ. 0.0) X(5) = X(5) + .1
      SLOPE=(Y(5)-Y(4))/(X(5)-X(4))
      Y(I)=((7.196-(APRESS(I)**.2857))*720.71)
      X(I)=(((Y(I)-Y(5)))/SLOPE)+X(5)
X     TYPE "Y(I)=",Y(I),"X(I)=",X(I)
      ATEMP(I)=((X(I)-500.0)/25.0)-70.0
C*****PERFORMS THE DEWPOINT INTERPOLATION.
      X(4)=(((TEMP1-DEPR1+70.0)*25.0)+500.0)
      X(5)=(((TEMP-DEPR+70.0)*25.0)+500.0)
      IF((X(5) - X(4)) .EQ. 0.0) X(5) = X(5) + .1
      SLOPE=(Y(5)-Y(4))/(X(5)-X(4))
X     TYPE "X(4)=",X(4),"X(5)=",X(5),"Y(4)=",Y(4),"Y(5)=",Y(5)
X     TYPE "SLOPE=",SLOPE
      X(I)=((Y(I)-Y(5))/SLOPE)+X(5)
X     TYPE "X(I)=",X(I)
      DEW(I)=((X(I)-500.0)/25.0)-70.0
X     TYPE "I= ",I," ATEMP(I)= ",ATEMP(I)," DEW(I)= ",DEW(I)
  628 I=I-1
      IF(I .EQ. 0) GO TO 650
      GO TO 622
```

```
C       COMPUTES THE K INDEX
C*****CALCULATES THE K INDEX.
  650 AK=-ATEMP(1)+ATEMP(3)+DEW(3)-(ATEMP(2)-DEW(2))+.5
      K=IFIX(AK)
      GO TO 656
  655 K=-1000
  656 IF(K .EQ. -1000) GO TO 675
C*****PUTS THE K VALUE INTO THE PLOT.
      IT(1)="K="
      IT(2)=" "
      IF(K .LT. 0) IT(2)="-"
      K=IABS(K)
      IT(3)=((K/10)*256)+12288
      IT(4)=((K-((K/10)*10))*256)+12288
      IT(5)=0
      GO TO 685
  675 IT(1)="K="
      IT(2)=" "
      IT(3)="NA"
      IT(4)=0
  685 CALL CLOSE(6,IER)
      CALL TEXT(NSIZE,IGA,IL,JL,IT,NW,NS)
C*****CREATES TEMPORARY SCRATCH FILES SO DATA CAN BE PASSED ON TO
C*****THE SECOND HALF OF THE PROGRAM.
      CALL DFILW("DP0:SCRATCH.T",IER)
      CALL CFILW("DP0:SCRATCH.T",2,IER)
      CALL OPEN(15,"DP0:SCRATCH.T",2,IER)
C*****WRITES OFF THE IGA ARRAY FOR RETENTION.
      CALL RDBLK(15,0,IGA,2,IER)
      CALL CLOSE(15,IER)
      CALL DFILW("DP0:SCRATCH.NW",IER)
      CALL CFILW("DP0:SCRATCH.NW",2,IER)
      CALL OPEN(16,"DP0:SCRATCH.NW",2,IER)
C*****WRITES OFF NW AND THE PRESSURE OF THE SURFACE LEVEL FOR RETENTION.
      WRITE(16,501) NW, SFCPRS
  501 FORMAT(1X,I3,1X,F5.0)
      CALL CLOSE(16,IER)
X     WRITE(10,700) NW,N
X 700 FORMAT(1X,"I MADE IT THRU PLOTRAOB1  NW = ",I4,"N = ",I4)
      CALL SWAP("PLOTRAOB2.SV",IER)
      CALL BACK
      STOP
      END
```

```
TYPE PLOTRAOB2.FR
      PARAMETER NS=512,NS2=1024
      DIMENSION IGA(NS),IB(NS2),IT(25),IVEC(10),IINT(10)
      CALL OPEN(15,"DP0:SCRATCH.T",2,IER)
C*****READS IN THE IGA ARRAY BEGUN IN PART 1.
      CALL WRBLK(15,0,IGA,2,IER)
      CALL CLOSE(15,IER)
      CALL DFILW("DP0:SCRATCH.T",IER)
      CALL OPEN(16,"DP0:SCRATCH.NW",2,IER)
C*****READS IN NW AND THE SURFACE PRESSURE LEVEL FROM PART 1.
      READ(16,502) NW, SFCPRS
  502 FORMAT(I3,1X,F5.0)
      CALL CLOSE(16,IER)
      CALL DFILW("DP0:SCRATCH.NW",2,IER)
C*****AN ARRAY ASSIGNING U.S. STANDARD ATMOSPHERE HEIGHTS TO THE
C*****PRESSURE LEVELS AT 100MB DECREMENTS.
      IINT(1)=04
      IINT(2)=32
      IINT(3)=64
      IINT(4)=99
      IINT(5)=138
      IINT(6)=184
      IINT(7)=236
      IINT(8)=300
      IINT(9)=386
      IINT(10)=532
      CALL OPEN(8,"SOUNDING.W",2,IER)
C*****READS IN THE HEIGHT AND IT'S WIND SPEED AND DIRECTION.
      READ(8,12) ISTATN,IDATE,ITIME
   12 FORMAT(I3,1X,I2,1X,I2)
  400 READ(8,401,END=300)ILVL,IDIR,ISPEED
  401 FORMAT(I2,1X,I3,I2)
C*****ASSIGNS THE HEIGHT LEVEL A PRESSURE VALUE, THEN A Y-COORDINATE
C*****LOCATION ON THE PLOT.
      ILVL=ILVL*10
      IF(ILVL .LT. IINT(1)) PRESS=1010.0
      IF(ILVL .EQ. 0) PRESS=SFCPRS
      IF(ILVL .LT. IINT(1)) GO TO 450
      DO 425 IJ=2,10
      IJM1=IJ-1
      A= FLOAT(ILVL) - FLOAT(IINT(IJM1))
      B=FLOAT(IINT(IJ)) - FLOAT(IINT(IJM1))
      A=(A/B)*100.0
      B=1100. - ((FLOAT(IJM1))*100.0)
      IF(ILVL .LT. IINT(IJ)) PRESS= B - A
      IF(ILVL .LT. IINT(IJ)) GO TO 450
  425 CONTINUE
  450 NP1=IFIX(((((7.196-PRESS**.2857)*720.71)+400.0))
      IVEC(1)=3800
      IVEC(2)=NP1
C*****ASSIGNS THE POINT FOR THE TAIL OF THE WIND VECTOR (ARROW).
      IDIR1=((IDIR/10)*10)
      IDIR5 = IDIR + 5
      IDIR51 = ((IDIR5/10)*10)
      IF(IDIR .NE. IDIR1 .AND. IDIR5 .NE. IDIR51) ISPEED=ISPEED+100
      ANGLE=((360.-(FLOAT(IDIR)))+90.)/57.29577
      AJ=150.*SIN(ANGLE)
      JL=(IFIX(AJ))+NP1
      AI=150.*COS(ANGLE)
      IL=(IFIX(AI))+3800
      IVEC(3)=IL
      IVEC(4)=JL
      IVEC(5)=999
```

```
C*****DRAWS THE WIND VECTOR ON THE PLOT.
      CALL MOVEVEC(IGA,IVEC,NW,NS)
C*****ASSIGNS THE WIND SPEED AND IT'S LOCATION ON THE PLOT.
      NSIZE=1
      IL=IL-100
      IF(IDIR .LT. 180) IL=IL+110
      IHUND=ISPEED/100
      IT(1)=(IHUND*256)+12288
      IHUND=IHUND*100
      ITEN=(ISPEED-IHUND)/10
      IT(2)=(ITEN*256)+12288
      ITEN=ITEN*10
      IONE=(ISPEED-IHUND-ITEN)
      IT(3)=(IONE*256)+12288
      IT(4)=0
X     TYPE NSIZE,IL,JL,NW
C*****PUTS THE WIND SPEED INTO THE PLOT.
      CALL TEXT(NSIZE,IGA,IL,JL,IT,NW,NS)
C*****TEST TO SEE IF NW WILL EXCEED DIMENSION ALLOWANCES (NS)
C*****IF SO,  IT GOES TO THE END.  NS CAN NOT BE LARGER TO AVOID
C*****THIS AND STILL HAVE THE PROGRAM RUN IN 10K MEMORY.
      IF(NW .GT. 472) GO TO 300
      GO TO 400
  300 NSIZE=3
C*****PUTS THE STATION NUMBER, DATE AND TIME INTO THE PLOT.
      IL=850
      JL=100
      IT(1)="ST"
      IT(2)="AT"
      IT(3)="IO"
      IT(4)="N="
      IHUND=ISTATN/100
      IT(5)=(IHUND*256)+12288
      IHUND=100*IHUND
      ITEN=(ISTATN-IHUND)/10
      IT(6)=(ITEN*256)+12288
      ITEN=10*ITEN
      IT(7)=((ISTATN-IHUND-ITEN)*256)+12288
      IT(8)="  "
      IT(9)=" "
      IT(10)="DA"
      IT(11)="TE"
      IT(12)="="
      ITEN=IDATE/10
      IT(13)=(ITEN*256)+12288
      ITEN=10*ITEN
      IT(14)=((IDATE-ITEN)*256)+12288
      IT(15)="  "
      IT(16)=" "
      ITEN=ITIME/10
      IT(17)="TI"
      IT(18)="ME"
      IT(19)="="
      IT(20)=(ITEN*256)+12288
      ITEN=10*ITEN
      IT(21)=((ITIME-ITEN)*256)+12288
      IT(22)="Z"
      IT(23)=0
X     TYPE NSIZE,IL,JL,NW
      CALL TEXT(NSIZE,IGA,IL,JL,IT,NW,NS)
      CALL CLOSE(8,IER)
X     TYPE NW
C*****PUTS THE PLOT INTO FINAL GRAPHIC FORM FOR DISPLAY.
      CALL UTF("RAOBPLOT",IGA,IB,NW,NS,NS2)
      CALL BACK
      STOP
      END
```

```
TYPE INDEXES.FR
C*****CALCULATES THE K INDEX, SHOWALTER INDEX, LIFTED INDEX, LIFTED
C*****CONDENSATION LEVEL AND IT'S TEMPERATURE, LEVEL OF FREE CONVECTION,
C*****CONVECTIVE CONDENSATION LEVEL AND IT'S TEMPERATURE, SURFACE
C*****TEMPERATURE AT WHICH CONVECTION WILL BEGIN, AND THE ENERGY IN THE
C*****POSITIVE AREA OF THE SOUNDING.
      DIMENSION PRESS(30),TEMP(30),DEPR(30),PRS(3),TMP(3),DEW(3),
     1TPCL(2)
      PRS(1)=850
      PRS(2)=700
      PRS(3)=500
C*****II OF 0 INDICATES THIS CALCULATION IS THE SHOWALTER INDEX.
      II=0
      CALL OPEN(20,"SOUNDING.T",2,IER)
      READ(20,20) ISTATN,IDATE,ITIME
   20 FORMAT(I3,1X,I2,1X,I2)
      N=1
C*****READS IN THE DATA FROM SOUNDING.T.
   25 READ(20,30,END=60) PRESS(N),TEMP(N),DEPR(N)
   30 FORMAT(F4.0,1X,F5.1,1X,F4.1)
      N=N+1
      IF(N .EQ. 31) GO TO 60
      GO TO 25
   60 CALL CLOSE(20,IER)
C*****INTERPOLATES FOR TEMPERATURE AND DEWPOINT VALUES AT 850, 700
C*****AND 500MB.
      DO 70 I=1,3
      CALL INTRPOLATE(PRS(I),PRESS,TEMP,DEPR,TMP(I),DEW(I),2)
   70 CONTINUE
C*****CHECK TO SEE IF THE NECESSARY MOISTURE DATA IS AVAILABLE.
      IF(DEW(1) .EQ. -99) TYPE "850 MOISTURE NOT AVAILABLE"
      IF(DEW(1) .EQ. -99) LFTIND=-99
      IF(DEW(1) .EQ. -99) ISHWLT =-99
      IF(DEW(1) .EQ. -99) GO TO 145
C*****TAKES 850MB TEMPERATURE AND DEWPOINT AND RAISES IT TO
C*****SATURATION(LCL).
      TD=DEW(1)+273.16
      T=TMP(1)+273.16
   50 TLCL=TD-(.001296*TD-.15772)*(T-TD)
      THETA=T*((1000./PRS(1))**.286)
      PLCL=1000.*((TLCL/THETA)**3.496)
      PRS(1)=850.
X     TYPE T,TD,PLCL,TLCL, "LAST TIME"
      IF(PLCL .LE. 500.) GO TO 120
C*****LIFTS THE SATURATED PARCEL TO 500MB.
      CALL LIFT(PLCL,TLCL,PRS(3),TF)
      GO TO 130
C*****SHOWALTER AND L.I. COMPUTATION, IF THE PARCEL IS UNSATURATED
C*****AT 500MB.
  120 T=THETA/(2.**.286)
      TF=T-273.16
      SHWLT=TMP(3)-TF
      GO TO 135
C*****SHOWALTER AND L.I. COMPUTATION, IF PARCEL IS SATURATED AT 500MB.
  130 SHWLT=TMP(3)-(TF-273.16)
  135 IF(II .EQ. 1) GO TO 150
C*****ROUNDS THE SHOWALTER INDEX TO AN INTEGER.
      CALL MKRND(SHWLT,ISHWLT)
C*****COMPUTES A MEAN LOW LEVEL DEWPOINT FOR THE LOWEST 50MB.
      CALL MMXRT(PRESS,TEMP,DEPR,TD)
      TD=TD+273.16
```

```
C*****CALCULATES AN ESTIMATED MAXIMUM TEMPERATURE FOR THE DAY.
      T7=TMP(2)+2.0+273.16
      THETA=T7*((1000./700.)**0.286)
      T=THETA/((1000./PRESS(1))**0.286)
C*****II OF 1 INDICATES THIS CALCULATION IS FOR THE L.I.
      II=1
      PRS(1)=PRESS(1)
      GO TO 50
  150 CALL MKRND(SHWLT,LFTIND)
X     TYPE "LIFTED INDEX = ",LFTIND
X     TYPE " SHOWALTER INDEX = ",ISHWLT
C*****CALCULATES THE K INDEX.
  145 IF(DEW(1) .EQ. -99 .OR. DEW(2) .EQ. -99) GO TO 155
      K=IFIX(TMP(1)-TMP(3)+DEW(1)-TMP(2)+DEW(2))
      GO TO 160
  155 K=-99
  160 CONTINUE
X     TYPE "K= ",K
C*****BEGINS EFFORTS TO CALCULATE THE CCL AND IT'S TEMPERATURE.
C*****COMPUTES MIXING RATIO VALUE USING THE MEAN LOW LEVEL DEWPOINT
X     TYPE "THE MEAN TD USED IN THE CCL IS ",TD
      EE=6.11*(273.16/TD)**5.31*EXP(25.22*(1.0-(273.16/TD)))
      WMXRT=(0.62197*EE)/(PRESS(1)-EE)
      ZPRES=PRESS(1)
  250 N=1
C*****DECREMENTS THE PRESSURE BY 10MB.
      ZPRES=ZPRES-10.0
  255 IF(ZPRES .LT. PRESS(N) .AND. ZPRES .GT. PRESS(N+1)) GO TO 260
      IF(ZPRES .EQ. PRESS(N)) GO TO 262
      IF(ZPRES .EQ. PRESS(N+1)) GO TO 263
      N=N+1
      GO TO 255
C*****FINDS A TEMPERATURE FOR THE NEW PRESSURE LEVEL.
  260 CALL INTRPOLATE(ZPRES,PRESS,TEMP,DEPR,TMPLVL,DEWLVL,1)
      T=TMPLVL
      GO TO 265
  262 T=TEMP(N)
      GO TO 265
  263 T=TEMP(N+1)
  265 EEI=ZPRES/(1.0+(0.62197/WMXRT))
      TDMX=1.0/(.003661-(.0001844*ALOG(EEI/6.11)))
      TDMX=TDMX-273.16
C*****COMPARES PARCEL TEMPERATURE WITH ENVIRONMENT TEMPERATURE.
      COMPR=T-TDMX
      IF(COMPR .EQ. 0.0) GO TO 310
      IF(COMPR .LT. 0.0) GO TO 290
      FPRES=ZPRES
      FCOMP=COMPR
      GO TO 250
C*****FINDS EXACT CROSSING OF PARCEL PATH WITH TEMPERATURE SOUNDING.
  290 SCOMP=ABS(COMPR)
      CCL=FPRES-((10.0*FCOMP)/(FCOMP+SCOMP))
      GO TO 220
  310 CCL=ZPRES
C*****ICCL IS THE CCL PRESSURE LEVEL AND TCCL IS IT'S TEMPERATURE
C*****IN DEGREES C.
  220 CALL MKRND(CCL,ICCL)
      CALL INTRPOLATE(CCL,PRESS,TEMP,DEPR,TCCL,DEWLVL,1)
X     TYPE "THE TCCL IS ",TCCL
```

```
C*****BRINGS THE PARCEL AT THE CCL TO THE SURFACE TO SEE WHAT SURFACE
C*****TEMPERATURE IS NECESSARY TO BEGIN CONVECTION (TCONV).
      THETA=(TCCL+273.16)*((1000./CCL)**.286)
      TCONV=(THETA/((1000./PRESS(1))**0.286))-273.16
X     TYPE "THE CCL = ",ICCL," CONV TEMP = ",TCONV
C*****CALLS A SUBROUTINE TO CALCULATE THE LCL AND IT'S TEMPERATURE,
C*****AND THE PRESSURE AND TEMPERATURE OF THE LIFTED CONDENSATION LEVEL.
      CALL LFC(TEMP,PRESS,DEPR,ILFC,LCLTMP,LCLPRS)
X     TYPE "LCL IS AT ",LCLPRS,"LCL TEMP IS ",LCLTMP
C*****CALCULATES THE POSITIVE AREA IN THE SOUNDING (ENERGY IN IT IN
C*****JOULES/GRAM).  THE POSITIVE AREA IS DEFINED BY LIFTING THE PARCEL
C*****AT THE CCL UNTIL IT EVENTUALLY BECOMES COOLER THAN THE ENVIRONMENT.
      TAREA=0.0
      PRS(1)=CCL
      TMP(1)=TCCL
      TPCL(1)=TCCL
      PRS(2) = FLOAT((ICCL/10)*10)
      TCCL=TCCL+273.16
  200 CALL INTRPOLATE(PRS(2),PRESS,TEMP,DEPR,TMP(2),DEWLVL,1)
X     TYPE "LIFT PARAMETERS",CCL,TCCL,PRS(2)
      CALL LIFT(CCL,TCCL,PRS(2),TPCL(2))
      TPCL(2)=TPCL(2)-273.16
X     TYPE "TPCL(2) IS",TPCL(2)
      COMPR=TPCL(2) - TMP(2)
      IF(COMPR .LT. 0.0) GO TO 205
      FCOMPR=COMPR
  206 TP=(TPCL(1)+TPCL(2))/2.0
      TE=(TMP(1)+TMP(2))/2.0
      TAREA=TAREA+((.287*(TP-TE))*(ALOG(PRS(1)/PRS(2))))
      IF(COMPR .LE. 0.0) GO TO 210
      PRS(1)=PRS(2)
      TPCL(1)=TPCL(2)
      TMP(1)=TMP(2)
      PRS(2)=PRS(2)-10.0
      GO TO 200
  205 SCOMP=ABS(COMPR)
      PRS(2) = PRS(1) - ((10.0*FCOMP)/(FCOMP + SCOMP))
C      TAREA IS THE POTENTIAL ENERGY IN THE SOUNDING IN JOULES/GRAM
      CALL INTRPOLATE(PRS(2),PRESS,TEMP,DEPR,TMP(2),DEWLVL,1)
      TPCL(2)=TMP(2)
      GO TO 206
  210 TCCL=TCCL-273.16
C*****CREATES A FILE "INDEXOUT" FOR THE OUTPUT INFORMATION.
      CALL DFILW("INDEXOUT",IER)
      CALL CFILW("INDEXOUT",2,IER)
      CALL OPEN(23,"INDEXOUT",2,IER)
C*****WRITES OUT THE OUTPUT.
      WRITE(23,524) ISTATN,IDATE,ITIME
      WRITE(23,525) K
      WRITE(23,526) ISHWLT
      WRITE(23,527) LFTIND
      WRITE(23,528) LCLPRS
      WRITE(23,529) LCLTMP
      IF(ILFC .EQ. -99) WRITE(23,535)
  524 FORMAT(6X,"STATION = ",I3,"   DATE = ",I2,"   TIME = ",I2,//)
  535 FORMAT(4X,"NO LEVEL OF FREE CONVECTION EXISTS BELOW 500MB",/)
      IF(ILFC .EQ. -99) GO TO 540
      WRITE(23,530) ILFC
  540 WRITE(23,531) ICCL
      CALL MKRND(TCCL,ITCCL)
      WRITE(23,532) ITCCL
      CALL MKRND(TCONV,ITCONV)
      WRITE(23,533) ITCONV
      WRITE(23,534) TAREA
  525 FORMAT(4X,"THE K INDEX = ",I3,//)
  526 FORMAT(4X,"THE SHOWALTER INDEX = ",I3,//)
  527 FORMAT(4X,"THE LIFTED INDEX = ",I3,//)
  528 FORMAT(4X,"THE LIFTED CONDENSATION LEVEL IS AT ",I4,"MB")
  529 FORMAT(4X,"AND HAS A TEMPERATURE OF ",I3," DEGREES C",//)
  530 FORMAT(4X,"THE LEVEL OF FREE CONVECTION IS AT ",I4,"MB",//)
  531 FORMAT(4X,"THE CONVECTIVE CONDENSATION LEVEL IS AT ",I4,"MB")
  532 FORMAT(4X,"AND HAS A TEMPERATURE OF ",I3," DEGREES C",//)
  533 FORMAT(4X,"CONVECTION WILL BEGIN WHEN THE SURFACE TEMPERATURE
     1 REACHES ",I3," DEGREES C",/)
  534 FORMAT(4X,"THE ENERGY IN THE POSITIVE AREA OF THE RAOB IS
     1 ",F5.3,"JOULES/GRAM")
X     TYPE "THE POS AREA IS ",TAREA
      STOP
      END
R
```

```
TYPE MKRND.FR
C*****ROUNDS OFF A POSITIVE OR NEGATIVE REAL NUMBER TO AN INTEGER VALUE.
      SUBROUTINE MKRND(BINT,IBINT)
      IBINT=IFIX(BINT)
      DEC=ABS(IBINT-BINT)
      IF(BINT.LT. 0.0)GO TO 30
      IF(DEC.GE.0.5)IBINT=IBINT+1
      GO TO 50
 30   IF(DEC.GE.0.5)IBINT=IBINT-1
 50   RETURN
      END
R
```

```
TYPE MMXRT.FR
C*****CALCULATES THE MIXING RATIO AT A LEVEL 50MB ABOVE THE SURFACE
C*****THEN FINDS AN AVERAGE DEWPOINT FOR THE LOWEST 50MB.
      SUBROUTINE MMXRT(PRESS,TEMP,DEPR,TD)
      DIMENSION PRESS(30),TEMP(30),DEPR(30)
      IF(DEPR(1).EQ.-1.0)GO TO 14
      TDMA=TEMP(1)-DEPR(1)+273.16
X     TYPE "TDMA =",TDMA,TEMP(1),DEPR(1)
      EEEA=6.11*(273.16/TDMA)**5.31*EXP(25.22*(1.0-
     *(273.16/TDMA)))
      XMXRA=(0.62197*EEEA)/(PRESS(1)-EEEA)
      I=2
  11  NPRES=PRESS(I)
      IF(NPRES.LT.(PRESS(1)-50.0))GO TO 13
      I=I+1
      GO TO 11
  13  TDMB=TEMP(I)-DEPR(I)+273.16
X     TYPE "TDMB =",TDMB,TEMP(I),DEPR(I)
      EEEB=6.11*(273.16/TDMB)**5.31*EXP(25.22*(1.0-
     *(273.16/TDMB)))
      XMXRB=(0.62197*EEEB)/(PRESS(I)-EEEB)
      PRSB=PRESS(1)-50.0
      XMXRC=(XMXRA*(PRESS(I)-PRSB)+XMXRB*(PRSB-PRESS(1)))
     */(PRESS(I)-PRESS(1))
      EEEC=PRESS(1)/(1.0+(0.62197/XMXRC))
      TDMC=1.0/(0.003661-(0.0001844*ALOG(EEEC/6.11)))
      TD=TDMC-273.16
      GO TO 17
C*****IF THE MOISTURE DATA IS MISSING, TD IS RETURNED AS -99.
  14  TD=-99
  17  RETURN
      END
R
```

```
TYPE INTRPOLATE.FR
      SUBROUTINE INTRPOLATE(PRSLVL,PRESS,TEMP,DEPR,TMPLVL,DEWLVL,IPART)
C*****INTERPOLATES BETWEEN PRESSURE LEVELS OF THE FILE SOUNDING.T TO
C*****GIVE THE TEMPERATURE AND DEW POINT AT ANY GIVEN PRESSURE
C*****LEVEL (PRSLVL).  A TEMPERATURE OR DEWPOINT OF -99 FROM THIS
C*****SUBROUTINE MEANS THAT NO INTERPOLATION WAS POSSIBLE OR THAT THE
C*****DEWPOINT WAS NOT MEASURED AT ONE OF THE LEVELS NESSESSARY FOR
C*****THE INTERPOLATION.  IF IPART IS 1 ONLY A TEMPERATURE INTERPOLATION
C*****IS PERFORMED.
      DIMENSION PRESS(30),TEMP(30),DEPR(30),Y(3),X(3)
C*****SEARCHES FOR THE PRESSURE LEVELS ON BOTH SIDES OF THE DESIRED LEVEL.
      J=1
      IF(PRSLVL .GT. PRESS(1)) GO TO 70
      IF(PRSLVL .EQ. PRESS(1)) GO TO 50
      PRESS1=PRESS(1)
      DO 20 I=2,30
      J=I
      IF(PRSLVL .EQ. PRESS(I)) GO TO 50
      IF(PRSLVL .GT. PRESS(I) .AND. PRSLVL .LT. PRESS1) GO TO 60
      PRESS1=PRESS(I)
      IF(I .EQ. 30) GO TO 70
   20 CONTINUE
C*****INTERPOLATES FOR THE TEMPERATURE.
   60 Y(1)=((7.196-(PRESS1**.2857))*720.71)
      Y(2)=((7.196-(PRESS(J)**.2857))*720.71)
      X(1)=(((TEMP(J-1)+70.0)*25.0)+500.0)
      X(2)=(((TEMP(J)+70.0)*25.0)+500.0)
      IF((X(2)-X(1)) .EQ. 0.0) X(2)=X(2)+.1
      SLOPE=(Y(2)-Y(1))/(X(2)-X(1))
      Y(3)=((7.196-(PRSLVL**.2857))*720.71)
      X(3)=((Y(3)-Y(2))/SLOPE)+X(2)
      TMPLVL=((X(3)-500.0)/25.0)-70.0
      IF(IPART .EQ. 1) GO TO 55
      IF(DEPR(J-1) .EQ. -1 .OR. DEPR(J) .EQ. -1) DEWLVL=-99
      IF(DEWLVL .EQ. -99) GO TO 100
C*****INTERPOLATES FOR THE DEWPOINT.
      X(1)=(((TEMP(J-1) - DEPR(J-1)+70.0)*25.0)+500.0)
      X(2)=(((TEMP(J)-DEPR(J)+70.0)*25.0)+500.0)
      IF((X(2)-X(1)) .EQ. 0.0) X(2)=X(2)+.1
      SLOPE=(Y(2)-Y(1))/(X(2)-X(1))
      X(3)=((Y(3)-Y(2))/SLOPE)+X(2)
      DEWLVL=((X(3)-500.0)/25.0)-70.0
      GO TO 100
   50 TMPLVL = TEMP(J)
   55 IF(IPART .EQ. 1) DEWLVL = -99
      IF(IPART .EQ. 1) GO TO 100
      DEWLVL = TEMP(J) - DEPR(J)
      GO TO 100
   70 TMPLVL = -99
      DEWLVL = -99
  100 RETURN
      END
R
```

```
TYPE LIFT.FR
C*****A MOIST ADIABATIC LIFT ROUTINE.
C*****GIVEN THE INITIAL PRESSURE (PI) AND TEMPERATURE (TI) OF A PARCEL
C*****AND THE LEVEL TO WHICH IT IS TO BE RAISED (PF), THIS GIVES THE
C*****FINAL TEMPERATURE (TF).
      SUBROUTINE LIFT(PI,TI,PF,TF)
      REAL L
      CPW=1.841
      EP=.622
      T=TI
      P=PI
      C=4.184
      CP=1.005
      R=.28704
10    CONTINUE
      E=6.11*((273.16/T)**5.31)*EXP(25.22*(1-273.16/T))
      TW=.622*E/(P-E)
      S=(CP+TW*C)*ALOG(T)-R*ALOG(P-E)+TW*(2499.1-2.343*(T-273.16))/T
      P=P-5.
      IF(P.LT.PF) P=PF
20    E=6.11*((273.16/T)**5.31)*EXP(25.22*(1-273.16/T))
      W=EP*E/(P-E)
      L=2499.1-2.343*(T-273.16)
      SK=(CP+TW*C)*ALOG(T)-R*ALOG(P-E)+W*L/T
      DS=(CP+TW*C-W*(C-CPW))/T+W*(EP+W)*L**2./(R*T**3.)
      T=T+(S-SK)/DS
      IF((S-SK).LT.1.E-7) GO TO 100
      GO TO 20
100   CONTINUE
      IF(P.EQ.PF) GO TO 110
      GO TO 10
110   CONTINUE
      TF=T
      RETURN
      END
```

```
TYPE LFC.FR
C*****THIS PROGRAM COMPUTES THE LEVEL OF FREE CONVECTION (LFC)
C*****BY FINDING THE LCL (USING THE SURFACE TEMPERATURE AND MEAN
C*****DEWPOINT OF THE LOWEST 50MB) THEN LIFTING MOIST ADIABATICALLY
C*****AND COMPARING WITH THE ACTUAL ENVIRONMENTAL TEMPERATURE.
      SUBROUTINE LFC(TEMP,PRESS,DEPR,ILFC,LCLTMP,LCLPRS)
      DIMENSION TEMP(30), PRESS(30)
      CALL MMXRT(PRESS,TEMP,DEPR,TD)
X     TYPE "TD =",TD
C*****FINDS THE LCL.
      I=0
      T=TEMP(1)+273.16
      TD=TD+273.16
      IF(T .GT. TD) GO TO 70
      TLCL = T
      PLCL = PRESS(1)
      GO TO 85
   70 TLCL=TD-(0.001296*TD-0.15772)*(T-TD)
      THETA=T*(1000.0/PRESS(1))**0.286
      PLCL=1000.0*(TLCL/THETA)**3.4965
      ZPRES=PLCL
   82 CALL INTRPOLATE(ZPRES,PRESS,TEMP,DEPR,T,DEWLVL,1)
      IF(I .EQ. 1) GO TO 85
      IF((TLCL-273.16) .GT. T) TYPE "LFC IS BELOW THE LCL"
      IF((TLCL-273.16) .GT. T) GO TO 150
  85  PI=PLCL
      TI=TLCL
      PF=ZPRES
      CALL LIFT(PI,TI,PF,TF)
X     TYPE "TF =",TF
      TDMX=TF-273.16
C*****COMPARES THE MOIST ADIABATIC VALUE WITH THE ACTUAL
C*****SOUNDING TEMPERATURE.
X     TYPE "T = ",T
      COMPR=T-TDMX
      IF(COMPR.EQ.0.0)GO TO 130
      IF(COMPR.LT.0.0)GO TO 110
      FPRES=ZPRES
      FCOMP=COMPR
      ZPRES=ZPRES-10.0
      IF(ZPRES.LT.500.0)GO TO 145
      I=1
      GO TO 82
 110  SCOMP=ABS(COMPR)
      ALFC=FPRES-((10.0*FCOMP)/(FCOMP+SCOMP))
      GO TO 140
 130  ALFC=ZPRES
 140  CALL MKRND(ALFC,ILFC)
X     TYPE"THE LEVEL OF FREE CONVECTION IS AT"
X     TYPE ILFC,"MB."
      GO TO 150
X     TYPE"NO LFC EXISTS BELOW 500MB."
X     TYPE "TF=",TF,"ZPRES=",ZPRES,"TLCL=",TLCL,"PLCL=",PLCL
C*****IF NO LFC EXISTS BELOW 500MB, ILFC IS RETURNED AS -99.
  145 ILFC=-99
C*****ASSIGNS THE LCL TEMPERATURE AND PRESSURE VALUES.
 150  LCLTMP=IFIX(TLCL-273.16)
      LCLPRS=IFIX(PLCL)
      RETURN
      END
R
```

```
TYPE WINDSHEAR.FR
C*****COMPUTES THE WINDSHEAR AND RICHARDSON NUMBER BETWEEN THE
C*****REPORTED LEVELS OF WIND.
      DIMENSION LEVEL(50),IDIRN(50),ISPEED(50),LVLA(5),LVLB(5),
     1IWSHR(5),RI(5),PRESS(30),TEMP(30)
      G=9.8
      CALL OPEN(5,"SOUNDING.W",2,IER)
      CALL OPEN(7,"SOUNDING.T",2,IER)
      I=1
C*****READS IN ALL THE DATA FROM THE FILE SOUNDING.T
      READ(7,10) ISTATN,IDATE,ITIME
    4 READ(7,5,END=16) PRESS(I),TEMP(I),DEPR
    5 FORMAT(F4.0,1X,F5.1,1X,F4.1)
      I=I+1
      GO TO 4
   16 CALL CLOSE(7,IER)
C*****READS IN ALL THE DATA FROM THE FILE SOUNDING.W
      READ(5,10) ISTATN,IDATE,ITIME
   10 FORMAT(I3,1X,I2,1X,I2)
      I=1
   15 READ(5,20,END=100) LEVEL(I),IDIRN(I),ISPEED(I)
   20 FORMAT(I2,1X,I3,I2)
      IF(((IDIRN(I)/5)*5) .EQ. IDIRN(I)) GO TO 25
      ISPEED(I)=ISPEED(I)+100
      IDIRN(I)=IDIRN(I)-1
   25 I=I+1
      GO TO 15
  100 I=I-2
C*****INITIALIZES ALL THE ARRAYS FOR THE FIVE GREATEST WINDSHEARS,
C*****THEIR LEVELS, AND THEIR RICHARDSON NUMBERS TO ZERO.
      DO 195 J=1,5
      LVLA(J)=0
      LVLB(J)=0
      IWSHR(J)=0
      RI(J)=0.0
  195 CONTINUE
C*****LOOP TO CALCULATE THE WINDSHEAR AND RICHARDSON NUMBER BETWEEN
C*****EACH REPORTED WIND
      DO 200 J=1,I
X     TYPE "J=",J
      IF(ISPEED(J) .LT. 10 .OR. ISPEED(J+1) .LT. 10) GO TO 200
C*****COMPUTES THE WINDSHEAR IN KNOTS/1000 FEET.
      CALL COMSHR(LEVEL,IDIRN,ISPEED,J,WNDSHR)
C*****ROUNDS THE WINDSHEAR OFF TO THE NEAREST INTEGER
      CALL MKRND(WNDSHR,IWNDSH)
X     TYPE WNDSHR
C*****ASSIGNS THE SURFACE HEIGHT LEVELS AT SLC TO 4000 FEET.  ALL
C*****OTHER STATIONS ARE ASSIGNED A SURFACE LEVEL OF 0 FEET (OR 1010MB
C*****SURFACE PRESSURE) IN THE SUBROUTINE PRSINT.
      IF(LEVEL(J) .EQ. 0 .AND. ISTATN .EQ. 572) LEVEL(J)=4
X     TYPE LEVEL(J)
C*****INTERPOLATES THE HEIGHT OF THIS WIND REPORT TO A PRESSURE
C*****VALUE IN THE U.S. STANDARD ATMOSPHERE.
      CALL PRSINT(LEVEL(J),PRSLVL)
X     TYPE PRSLVL
C*****INTERPOLATES AN ENVIRONMENTAL TEMPERATURE FOR THE DESIRED
C*****PRESSURE LEVEL.
      CALL INTRPOLATE(PRSLVL,PRESS,TEMP,DEPR,TMPLVL,DEWLVL,1)
      THETA1=(TMPLVL+273.16)*((1000./PRSLVL)**.286)
      CALL PRSINT(LEVEL(J+1),PRSLVL)
X     TYPE PRSLVL
      CALL INTRPOLATE(PRSLVL,PRESS,TEMP,DEPR,TMPLVL,DEWLVL,1)
X     TYPE TMPLVL
```

```
C*****COMPUTES VARIABLES FOR THE RICHARDSON NUMBER, WHICH IS RNUM.
      THETA2=(TMPLVL+273.16)*((1000./PRSLVL)**.286)
      THETAM=(THETA1+THETA2)/2.0
      DTHETA=THETA2-THETA1
      DZ=((LEVEL(J+1)-LEVEL(J))/3.2808399)*1000.
      IF(WNDSHR .EQ. 0.0) WNDSHR=.00001
      DVDZ2=((WNDSHR*3.2808399)/1942.54)**2
      RNUM=((G/THETAM)*(DTHETA/DZ))/DVDZ2
X     TYPE RNUM
      IF(IWNDSH .EQ. IWSHR(1))  GO TO 125
      IF(IWNDSH .LT. IWSHR(1)) GO TO 120
C*****A DOWNWARD SHIFT IN RANKING TO ALL WINDSHEARS LOWER OR EQUAL
C*****TO THIS COMPUTED VALUE IS PERFORMED HERE.
      CALL BUMP(LVLA,LVLB,IWSHR,RI,5)
      LVLA(1)=LEVEL(J)
      LVLB(1)=LEVEL(J+1)
      IWSHR(1)=IWNDSH
      RI(1)=RNUM
      GO TO 200
  120 IF(IWNDSH .EQ. IWSHR(2)) GO TO 145
      IF(IWNDSH .LT. IWSHR(2)) GO TO 140
  125 CALL BUMP(LVLA,LVLB,IWSHR,RI,4)
      LVLA(2)=LEVEL(J)
      LVLB(2)=LEVEL(J+1)
      IWSHR(2)=IWNDSH
      RI(2)=RNUM
      GO TO 200
  140 IF(IWNDSH .EQ. IWSHR(3)) GO TO 165
      IF(IWNDSH .LT. IWSHR(3)) GO TO 160
  145 CALL BUMP(LVLA,LVLB,IWSHR,RI,3)
      LVLA(3)=LEVEL(J)
      LVLB(3)=LEVEL(J+1)
      IWSHR(3)=IWNDSH
      RI(3)=RNUM
      GO TO 200
  160 IF(IWNDSH .EQ. IWSHR(4)) GO TO 185
      IF(IWNDSH .LT. IWSHR(4)) GO TO 180
  165 CALL BUMP(LVLA,LVLB,IWSHR,RI,2)
      LVLA(4)=LEVEL(J)
      LVLB(4)=LEVEL(J+1)
      IWSHR(4)=IWNDSH
      RI(4)=RNUM
      GO TO 200
  180 IF(IWNDSH .LE. IWSHR(5)) GO TO 200
  185 LVLA(5)=LEVEL(J)
      LVLB(5)=LEVEL(J+1)
      IWSHR(5)=IWNDSH
      RI(5)=RNUM
  200 CONTINUE
      CALL CLOSE(5,IER)
C*****OUTPUT FILE, "WINDSHEAR", IS CREATED AND THE OUTPUT IS
C*****PLACED IN IT.
      CALL DFILW("WINDSHEAR",IER)
      CALL CFILW("WINDSHEAR",2,IER)
      CALL OPEN(6,"WINDSHEAR",2,IER)
      WRITE(6,300)
  300 FORMAT(4X,"THE 5 LAYERS WITH THE STRONGEST WIND SHEAR,
     1 IN DECREASING")
      WRITE(6,301)
  301 FORMAT(2X,"ORDER, ALONG WITH THE LAYER'S RICHARDSON NUMBER ARE:"
     1,///)
      WRITE(6,302)
  302 FORMAT(7X,"LAYER",7X,"SHEAR",6X,"RICHARDSON")
      WRITE(6,303)
  303 FORMAT(5X,"1000'S FT",2X,"KNOTS/1000FT",4X,"NUMBER",///)
      DO 310 K=1,5
      WRITE(6,315) K,LVLA(K),LVLB(K),IWSHR(K),RI(K)
  315 FORMAT(2X,I1,3X,I2," - ",I2,7X,I2,9X,F7.2)
  310 CONTINUE
      STOP
      END
R
```

```
TYPE COMSHR.FR
      SUBROUTINE COMSHR(LEVEL,IDIRN,ISPEED,J,WNDSHR)
C*****GIVEN A LEVEL "J", COMPUTES THE VECTOR WIND SHEAR BETWEEN
C*****LEVEL(J) AND LEVEL(J+1)
      DIMENSION LEVEL(50), IDIRN(50),ISPEED(50)
      ADIR=IDIRN(J)*.01745329
      BDIR=IDIRN(J+1)*.01745329
      COSJ=(SIN(ADIR))*FLOAT(ISPEED(J))
      COSK=(SIN(BDIR))*FLOAT(ISPEED(J+1))
      IF(COSJ .LT. 0.0 .AND. COSK .GT. 0.0) GO TO 130
      IF(COSK .LT. 0.0 .AND. COSJ .GT. 0.0) GO TO 130
      DSPD1=ABS(COSJ-COSK)
      GO TO 140
  130 DSPD1=ABS(COSJ) + ABS(COSK)
  140 SHR1=DSPD1
      SINJ=(COS(ADIR))*FLOAT(ISPEED(J))
      SINK=(COS(BDIR))*FLOAT(ISPEED(J+1))
      IF(SINJ .LT. 0.0 .AND. SINK .GT. 0.0) GO TO 150
      IF(SINK .LT. 0.0 .AND. SINJ .GT. 0.0) GO TO 150
      DSPD2=ABS(SINJ-SINK)
      GO TO 160
  150 DSPD2=ABS(SINJ) + ABS(SINK)
  160 SHR2=DSPD2
      WNDSHR=SQRT((SHR1**2)+(SHR2**2))
X     TYPE "SHR1=",SHR1,"SHR2=",SHR2,"WNDSHR=",WNDSHR
C*****REDUCES THE WINDSHEAR TO A VALUE PER 1000 FEET.
      WNDSHR=WNDSHR/(FLOAT(LEVEL(J+1) - LEVEL(J)))
X     TYPE "WNDSHR=",WNDSHR
      RETURN
      END
R
```

```
TYPE PRSINT.FR
      SUBROUTINE PRSINT(IHT,PRSLVL)
C*****THIS IS ADMITTEDLY CRUDE..BUT IT WORKS.....
C*****INTERPOLATES A GIVEN HEIGHT LEVEL IN HUNDREDS OF FEET TO A
C*****PRESSURE LEVEL IN THE U.S. STANDARD ATMOSPHERE.
      DIMENSION IINT(10)
X     TYPE IHT
      IINT(1)=04
      IINT(2)=32
      IINT(3)=64
      IINT(4)=99
      IINT(5)=138
      IINT(6)=184
      IINT(7)=236
      IINT(8)=300
      IINT(9)=386
      IINT(10)=532
      ILVL=IHT*10
X     TYPE ILVL
      IF(ILVL .LT. IINT(1)) PRSLVL=1010.0
      IF(ILVL .LT. IINT(1)) GO TO 450
      DO 425 IJ=2,10
      IJM1=IJ-1
      IF(ILVL .GE. IINT(IJ)) GO TO 425
      A=FLOAT(ILVL)-FLOAT(IINT(IJM1))
      B=FLOAT(IINT(IJ))-FLOAT(IINT(IJM1))
      A=(A/B)*100.0
      B=1100. - ((FLOAT(IJM1))*100.0)
      PRSLVL=B-A
      GO TO 450
  425 CONTINUE
  450 RETURN
      END
R
```

```
TYPE BUMP.FR
      SUBROUTINE BUMP(LVLA,LVLB,IWSHR,RI,J)
C*****GIVEN THE LEVEL, "J", WHERE THE NEW VALUE IS TO GO, THIS PUSHES
C*****DOWN THE RANKING OF THE REST OF THE NUMBERS TO MAKE ROOM FOR IT.
      DIMENSION LVLA(5),LVLB(5),IWSHR(5),RI(5)
      DO 110 K=2,J
      L=7-K
      LM1=L-1
      LVLA(L)=LVLA(LM1)
      LVLB(L)=LVLB(LM1)
      IWSHR(L)=IWSHR(LM1)
      RI(L)=RI(LM1)
  110 CONTINUE
      RETURN
      END
R
```

APPENDIX

In order for these routines to be useful on AFOS, they need to be run and displayed on AFOS with as few operator commands as possible. The following are a description of the AFOS PROCEDURES which have been used to initiate the routines, and the PROCEDURES themselves.

1. To Select the Analysis Routines to be Performed on a Raob

   This is a simple task if the operator is able to use a PREFORMAT to interact with the AFOS system. The preformat, with all the raob routines listed on it, can be called up for examination. The operator can indicate which routines are to be run, and this can be stored in the data base as an AFOS file. The AFOS file can be SAVEd as an RDOS file, which can be read via an initiation program, and the selected routines run. The following is a description of such a PROCEDURE:

   a. The raob must be SAVEd as an RDOS file named "RAOB.BB" (i.e., SAVE: SLCSGLSLC DP1:RAOB.BB).

   b. The raob routine selection PREFORMAT must be called up.

   c. The header block should be filled out with some predesignated key (in the example WRHGENUSE is used).

   d. Tab down to the appropriate box and change the "Ø" to "1" for the routines that are to be run.

   e. Save the AFOS key (product) as an RDOS file named "CHECK".

   f. Run a program that reads the Ø's and 1's from "CHECK" and swaps in and out the programs to be run.

   g. STORE the output files as AFOS files if they are to be stored as part of the data base.

   h. Display the output files if desired.

The following is an AFOS PROCEDURE to do the above, the preformat needed, and a listing of the program "RUNNER" which is used to swap in and out the three routines currently available (the other three are planned for the future).

```
WRMPCD025
WOUS00 KWRH 241929
  DISPLAY    MODE  ACC/OV    COMMAND
   (1-4)     (D/I) (R/O/O)   (ANY COMMAND; LAST LINE MUST BE END OR "NONE")
  -------    ----- -------   ------------------------------------------------
01                             SAVE:SLCSGLSLC DP1:RAOB.CO
02                             PAUSE 00
03                             M:CO7
04                             PAUSE 5
05                             SAVE:WRHGENUSE DP1:CHECK
06                             PAUSE 00
07                             RUN:DP1:RUNNER.SV
08                             DELAY 3
09                             STORE:DP1:RAOBPLOT WRHRABSLC
10                             STORE:DP1:INDEXOUT WRHCONVEC
11                             STORE:DP1:WINDSHEAR WRHWINDSH
12                             END
13
14
15
16
17
18
```

```
WRHMCP007
WOUS00 KWRH 201819

CHANGE THE 0-S TO 1-S IN THE BOXES BESIDE THE ROUTINES YOU WANT TO RUN
ON THE CURRENT RAOB:

   VISUAL RAOB PLOT                [0]
   CONVECTIVE ANALYSIS             [0]
   ICING POTENTIAL ANALYSIS        [0]
   WIND SHEAR ANALYSIS             [0]
   TYPE OF PRECIPITATION ANALYSIS  [0]
   TEMPERATURE ADVECTION ANALYSIS  [0]

WHEN FINISHED MOVE CURSOR HERE [ ] AND HIT ENTER.
```

```
TYPE RUNNER.FR
      NR=4
      IMASKR=377K
      CALL SWAP("SGLDECODER.SV",IER)
      CALL OPEN(21,"CHECK",2,IER,1)
X     TYPE "I DID THE DECODE"
      NR=NR-1
      DO 10 I=0,NR,1
      ISPOT=173+(I*38)
      CALL READR(21,ISPOT,J,1,IER)
      J=ISHFT(J,-8)
      J=IAND(J,IMASKR)-060K
X     TYPE "J = ",J
      IF(I .EQ. 0 .AND. J .EQ. 1) CALL SWAP("PLOTRAOB1.SV",IER)
X     IF(I .EQ. 0 .AND. J .EQ. 1) TYPE "I DID PLOTRAOB1"
      IF(I .EQ. 1 .AND. J .EQ. 1) CALL SWAP("INDEXES.SV",IER)
X     IF(I .EQ. 1 .AND. J .EQ. 1) TYPE "I DID THE INDEXES"
      IF(I .EQ. 3 .AND. J .EQ. 1) CALL SWAP("WINDSHEAR.SV",IER)
X     IF(I .EQ. 3 .AND. J .EQ. 1) TYPE "I DID THE WINDSHEAR ANALYSIS"
   10 CONTINUE
      STOP
      END
R
```

2. To Plot 5 Different Raobs

This procedure allows the forecaster to enter a PREFORMAT and designate which five raobs in the Western Region are to be plotted (the subroutine "NDASSIGN" can be changed to allow raobs outside the Western Region to be recognized). There is an offset here that must be weighed by the user. In order to make this very flexible, yet involve few operator commands, a number of computer time consuming tasks must be used. It takes about 11 minutes for the 5 raobs to be selected, plotted, and stored into the data base. Hence, in demanding little operator input, the user is increasing the run time for the plots.

In this example a PREFORMAT is called up which allows the operator to input the five 3-letter designators for the raobs that are to be plotted. After that, no more operator intervention is needed.

The PREFORMAT information is SAVEd as an RDOS file named "TEST". This information is used to change another procedure (which runs the plots to contain these stations in it) then initiates this procedure to decode them and plot them. This is done as follows:

a. Call up the PREFORMAT.

b. Fill out the header block (in this example as WRHGENUSE).

c. Fill in the station call letters for the raobs to be plotted.

d. SAVE the AFOS file as an RDOS file named "TEST".

e. SAVE the procedure to plot 5 raobs (WRHPDC022 in this example) as an RDOS file named "RAOBPCD".

f. Execute the program "READER.SV" which reads the station call letters from "TEST", finds the WSFO node that the station call comes from, and substitutes the node and station names into "RAOBPCD".

g. STORE the changed file "RAOBPCD" back into the AFOS data base as a PROCEDURE.

h. Run the changed PROCEDURE (i.e., "Ø22" or "RAOB" in this example).

The procedure and preformat described above look like this:

```
WRHPCD039
WOUS00 KWRH 161457
  DISPLAY   MODE  ACC/OV    COMMAND
   (1-4)    (D/M) (R/A/O)   (ANY COMMAND; LAST LINE MUST BE END OR "NAME")
  -------   ----- -------   ----------------------------------------------
01                          M:006
02                          PAUSE 7
03                          SAVE:WRHTSTJAJ DP1:TEST
04                          PAUSE 7
05                          SAVE:WRHPCD022 DP1:RAOBPCD
06                          PAUSE 7
07                          RUN:DP1:READER.SV
08                          STORE:DP1:RAOBPCD WRHPCD022
09                          PAUSE 5
10                          "RAOB"
11                          END
12
13
14
15
16
17
18
```

PAGE 01

```
WRHMCP006
WOUS00 KWRH 141713

FILL IN THE BLANKS WITH THE CALL LETTERS OF THE RAOB STATIONS
THAT YOU WANT PLOTTED.  MAKE SURE THAT YOU HAVE THE NECESSARY KEYS
IN YOUR DATA BASE SO THAT THE PLOTTED SOUNDING FILE CAN BE STORED
(RAOBPHXXX, WHERE XXX IS THE STATION CALL LETTERS).  A MAXIMUM OF
FIVE STATION CALL LETTERS MAY BE ENTERED.

          [    ]
          [    ]
          [    ]
          [    ]
          [    ]
           950

 MOVE CURSOR HERE [  ], THEN HIT ENTER.
```

The Fortran programs which accomplish this are fairly straight forward. "READER.SV" uses Fortran read records to read the station call letters from the file "TEST". The station call letters (3 ASCII characters) are then left justified in the two word (two element) integer array that contains them via a subroutine named "LFTJST".

Next, a subroutine "NDASSIGN" is called. This is used to match the station name with its' node name, and return the node name to the main program. Another subroutine, "SETKEY", prepares and writes the station information into the proper locations of the procedure to plot the five raobs. The odd numbered assignments have to have the station name and node name right justified. This is done in a subroutine named "RGTJST". When this is done, the PROCEDURE is ready to be run.

The following is a listing of the main program "READER" and its' subroutine discussed above.

```
TYPE READER.FR
      DIMENSION ID(2), INODE(2)
      CALL OPEN(20,"TEST",2,IER,2)
      DO 10 I=1,5
      II=(180+((I-1)*7))
      CALL READR(20,II,ID,2,IER)
      CALL LFTJST(ID)
X     WRITE(10,25) ID(1), ID(2)
X  25 FORMAT(1X,2A2)
      CALL NDASSIGN(ID,INODE)
      CALL SETKEY(I,ID,INODE)
   10 CONTINUE
      CALL CLOSE(20,IER)
      STOP
      END
R
```

```
TYPE LFTJST.FR
      SUBROUTINE  LFTJST(ID)
      DIMENSION ID(2)
      IMASKR=377K
      IMASKL=177400K
      ID(1)=ISHFT(ID(1),8)
      ID(1)=IAND(ID(1),IMASKL)
      IRIGHT=ISHFT(ID(2),-8)
      IRIGHT=IAND(IMASKR,IRIGHT)
      ID(1)=IOR(ID(1),IRIGHT)
      ID(2)=ISHFT(ID(2),8)
      ID(2)=IAND(ID(2),IMASKL)
      RETURN
      END
R
```

```
TYPE RGTJST.FR
      SUBROUTINE RGTJST(ID)
      DIMENSION ID(2)
      IMASKR=377K
      IMASKL=177400K
      ID(2)=ISHFT(ID(2),-8)
      ID(2)=IAND(ID(2),IMASKR)
      ILEFT=ISHFT(ID(1),8)
      ILEFT=IAND(ILEFT,IMASKL)
      ID(2)=IOR(ILEFT,ID(2))
      ID(1)=ISHFT(ID(1),-8)
      ID(1)=IAND(IMASKR,ID(1))
      RETURN
      END
R
```

```
TYPE NDASSIGN.FR
      SUBROUTINE NDASSIGN(ID,INODE)
      DIMENSION ISTATN(42), ND(42), ID(2), INODE(2)
      IMASKL=177400K
      ISTATN(1)="AB"
      ISTATN(2)="Q "
      ISTATN(3)="BO"
      ISTATN(4)="I "
      ISTATN(5)="BI"
      ISTATN(6)="S "
      ISTATN(7)="LN"
      ISTATN(8)="D "
      ISTATN(9)="GJ"
      ISTATN(10)="T "
      ISTATN(11)="DE"
      ISTATN(12)="N "
      ISTATN(13)="GG"
      ISTATN(14)="W "
      ISTATN(15)="GT"
      ISTATN(16)="F "
      ISTATN(17)="SA"
      ISTATN(18)="N "
      ISTATN(19)="SL"
      ISTATN(20)="E "
      ISTATN(21)="MF"
      ISTATN(22)="R "
      ISTATN(23)="IN"
      ISTATN(24)="W "
      ISTATN(25)="TU"
      ISTATN(26)="S "
      ISTATN(27)="WM"
      ISTATN(28)="C "
      ISTATN(29)="EL"
      ISTATN(30)="Y "
      ISTATN(31)="DR"
      ISTATN(32)="A "
      ISTATN(33)="UI"
      ISTATN(34)="L "
      ISTATN(35)="GE"
      ISTATN(36)="G "
      ISTATN(37)="DA"
      ISTATN(38)="K "
      ISTATN(39)="UB"
      ISTATN(40)="G "
      ISTATN(41)="SL"
      ISTATN(42)="C "
      ND(1)="AB"
      ND(2)="Q "
      ND(3)="BO"
      ND(4)="I "
      ND(5)="BI"
      ND(6)="S "
      ND(7)="CY"
```

```
      ND(8)="S  "
      ND(9)="DE"
      ND(10)="N  "
      ND(11)="DE"
      ND(12)="N  "
      ND(13)="GT"
      ND(14)="F  "
      ND(15)="GT"
      ND(16)="F  "
      ND(17)="LA"
      ND(18)="X  "
      ND(19)="PD"
      ND(20)="X  "
      ND(21)="PD"
      ND(22)="X  "
      ND(23)="PH"
      ND(24)="X  "
      ND(25)="PH"
      ND(26)="X  "
      ND(27)="RN"
      ND(28)="0  "
      ND(29)="RN"
      ND(30)="0  "
      ND(31)="RN"
      ND(32)="0  "
      ND(33)="SE"
      ND(34)="A  "
      ND(35)="SE"
      ND(36)="A  "
      ND(37)="SF"
      ND(38)="0  "
      ND(39)="SF"
      ND(40)="0  "
      ND(41)="SL"
      ND(42)="C  "
      ID(2)=ID(2) + 040K
      N=0
      DO 50 J=1,41,2
      JP1=J+1
      IF(ID(1) .EQ. ISTATN(J) .AND. ID(2) .EQ. ISTATN(JP1)) N=J
      IF(N .EQ. J) GO TO 55
      IF(J .EQ. 41) WRITE(10,56) ID(1), ID(2)
   56 FORMAT(1X,"THE STATION ENTERED, ",2A2," DOES NOT MATCH ANY
     1UPPER AIR STATINS IN THIS PROGRAM")
   50 CONTINUE
      IF(N .EQ. 0) GO TO 60
   55 NP1=N+1
      INODE(1)=ND(N)
      INODE(2)=ND(NP1)
      INODE(2)=IAND(INODE(2),IMASKL)
      ID(2)=ID(2) - 040K
   60 RETURN
      END
R
```

```
TYPE SETKEY.FR
      SUBROUTINE SETKEY(I,ID,INODE)
      DIMENSION ID(2),INODE(2),IHEADG(5),ILOCA(5),ILOCB(5)
      IMASKR=377K
      IMASKL=177400K
      ILOCA(1)=25
      ILOCA(2)=162
      ILOCA(3)=298
      ILOCA(4)=435
      ILOCA(5)=571
      ILOCB(1)=152
      ILOCB(2)=289
      ILOCB(3)=425
      ILOCB(4)=562
      ILOCB(5)=698
      J=(I/2)*2
      IF(J .NE. I) GO TO 100
      CALL OPEN(21,"RAOBPCD",2,IER,2)
      IHEADG(1)=INODE(1)
      IHEADG(2)=" S"
      IHEADG(2)=IAND(IHEADG(2),IMASKR)
      IHEADG(2)=IOR(IHEADG(2),INODE(2))
      IHEADG(3)="GL"
      IHEADG(4)=ID(1)
      IHEADG(5)="  "
      IHEADG(5)=IAND(IHEADG(5),IMASKR)
      IHEADG(5)=IOR(IHEADG(5),ID(2))
      GO TO 110
  100 CALL RGTJST(INODE)
      CALL RGTJST(ID)
      CALL OPEN(21,"RAOBPCD",2,IER,2)
      IHEADG(1)=": "
      IHEADG(1)=IAND(IHEADG(1),IMASKL)
      IHEADG(1)=IOR(IHEADG(1),INODE(1))
      IHEADG(2)=INODE(2)
      IHEADG(3)="SG"
      IHEADG(4)="L "
      IHEADG(4)=IAND(IHEADG(4),IMASKL)
      IHEADG(4)=IOR(IHEADG(4),ID(1))
      IHEADG(5)=ID(2)
      IDUMMY="B "
      IDUMMY=IAND(IDUMMY,IMASKL)
      ID(1)=IOR(ID(1),IDUMMY)
  110 CALL WRITR(21,ILOCA(I),IHEADG,5,IER)
      CALL WRITR(21,ILOCB(I),ID,2,IER)
      CALL CLOSE(21,IER)
      RETURN
      END
R
```

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

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

The six Major Line Components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

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

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

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

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

TECHNICAL SERVICE PUBLICATIONS—These are publications containing data, observations, instructions, etc. A partial listing: Data serials; Prediction and outlook periodicals; Technical manuals, training papers, planning reports, and information serials; and Miscellaneous technical publications.

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

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER
ENVIRONMENTAL DATA SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

3300 Whitehaven Street, N.W.
Washington, D.C. 20235