PNL-5382
UC-95a

User's Guide for TRANZ: A Data Transformation and Analysis Program

K. J. Fischer

December 1985

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

Battelle

PNL-5382

DISCLAIMER

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

PNL-5382
UC-95a

USER'S GUIDE FOR TRANZ: A DATA TRANSFORMATION AND ANALYSIS PROGRAM

K. J. Fischer

December 1985

Prepared for
the Bonneville Power Administration
under a Related Services Agreement
with the U.S. Department of Energy
Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

FORWARD

This report is one in a series of documents describing particular elements of the End-Use Load and Conservation Assessment Program (ELCAP), which is being conducted by Pacific Northwest Laboratory (PNL) for the Bonneville Power Administration (BPA). The project's objective is to efficiently and accurately compile hourly end-use energy consumption levels and characteristic data for buildings and residences in the Pacific Northwest. These data will be used to characterize the energy requirements and to refine the accuracy of estimates of conservation potential in residential housing stock throughout the Pacific Northwest.

The project entails the auditing and metering of 700 residences and 250 buildings throughout the BPA service area during 1984 and 1985. The data will be configured into a data base management system to facilitate detailed analysis of building energy-use cause and effect. This report provides an insight into procedures used to ensure the quality of the data collected.

This report was prepared by Kenneth J. Fischer, a senior mechanical engineering student from Oregon State University, who participated in the summer intern program at PNL during 1984.

William F. Sandusky
ELCAP Installation Task Leader

SUMMARY

This report is a user's guide for TRANZ, a program developed by Pacific Northwest Laboratory (PNL) for an IBM PC® with a basic compiler, Version 2.0. The purpose of TRANZ is to convert, organize, and evaluate electrical end-use load data obtained from the Field Data Acquisition Systems (FDAS). These FDAS will be installed in commercial buildings and residences as part of the End-Use Load and Conservation Assessment Program (ELCAP) that is being managed by PNL for the Bonneville Power Administration (BPA). TRANZ has been developed with highly interactive menu-driven routines and requires little computer experience for proper operation. It is executable on a single or double disk drive IBM PC®. The program can be used for spot checking data, troubleshooting installation problems, and preparing tabular and graphical summaries. Sample files and program output, and a program source code listing are provided as appendices.

The program itself contains six primary routines: The first routine converts the raw character byte data dumped from the FDAS into a formatted ASCII engineering unit file. The second routine checks the data to make sure that each value is within a reasonable range. The third routine sorts a file so that the records start at the earliest time and end at the latest time. The fourth routine appends two files for the same building or residence, removing any overlap in the files. The fifth routine summarizes the data in both a tabular and graphical form. The sixth routine generates a hard copy table of day numbers and their corresponding dates.

® IBM PC is a trademark of International Business Machines, Corp., Boca Raton, Florida.

CONTENTS

1.0 INTRODUCTION

Pacific Northwest Laboratory (PNL) is managing the End-Use Load and Conservation Assessment Program (ELCAP) for the Bonneville Power Administration (BPA). To ensure the quality of the end-use data collected under the ELCAP effort, it is necessary to have techniques that can be used to support the data verification process. The verification process is required after the Field Data Acquisition System (FDAS) has been installed in commercial buildings and residences. This process determines whether or not the data being acquired is accurate and representative of actual conditions. One method that can be used in the verification process is to list and plot acquired data for various time intervals. It is desirable to have this process of either listing or plotting the data done independently of PNL's normal data acquisition and analysis tasks. To meet this objective, a Basic computer program, TRANZ, has been developed by PNL to operate on an IBM PC®. This document provides a user's guide for TRANZ.

The program was created under Disk Operating System (DOS) Version 2.0 and is menu driven to facilitate its use. The program consists of a main module and six primary routines. These routines deal with various aspects of data processing, summarization, and presentation. The latter can be either in the listing or plotting mode. A flow diagram of the program is illustrated in Figure 1.

Sample input files and program output, and a program source code listing are provided in Appendices A and B, respectively, to document correct operation of the program. The program is structured in such a manner that it can be easily converted to other PC types.

® IBM PC is a trademark of the International Business Machines, Corp., Boca Raton, Florida.

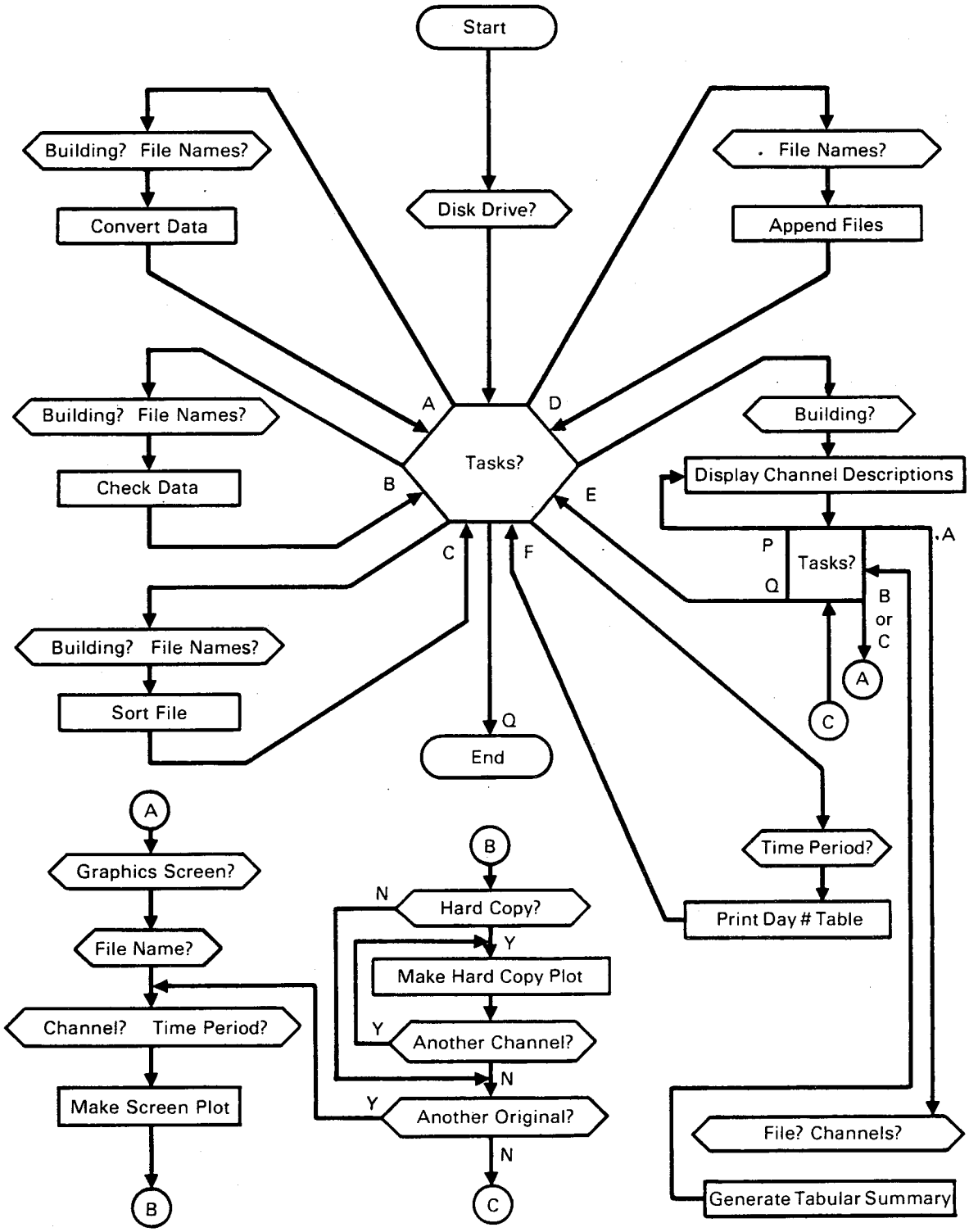

FIGURE 1. TRANZ Program Flow Diagram

2.0 PROGRAM CHARACTERISTICS

2.1 RUNTIME REQUIREMENTS

Before running TRANZ, check the program disk directory to make sure both TRANZ.EXE and BASRUN.EXE are included. The BASRUN.EXE file is required for the proper initialization of the compiled program. If both files are there, start the program by entering 'TRANZ' after a system prompt. To initialize the computer and start DOS, the user must have a system disk with the proper 'AUTOEXEC.BAT' file containing commands that are automatically executed when the system is initialized (an important process for the printer interface). If the user is unfamiliar with DOS, check the DOS manual and the printer manual for the proper 'AUTOEXEC.BAT' file. To initialize the system, place the disk in drive 'A:' while simultaneously pressing 'Ctrl', 'Alt', and 'Del'. If the computer is turned off, simply place the disk in drive 'A:' and turn the power on.

2.2 PROGRAM STRUCTURE

The first prompt given by the program is as follows:

ENTER DISK DRIVE FOR BUILDING DATA FILES (A:, B:, etc.)
DISK MUST CONTAIN '.DAT' AND '.PAR' FILES?

After entering the drive specification ('A:' or 'B:'), which tells where the file disk will be located, the program's main menu is displayed. Table 1 gives the six main routines for TRANZ.

To choose a routine, press the corresponding key (see Table 1). The first four routines ('A' through 'D') prepare the file so that it can be summarized using the fifth routine ('E'). Not every routine, however, must be used in all cases. All routines, except the summary routine, perform the desired function once for one building and one file, and they automatically return the user to the main menu. The summary routine has its own menu (see Section 7). Once the user enters the summary routine, summary work must be done for the same

TABLE 1. Example of Main Menu

Key	Choice
A	Convert raw data to an engineering (ASCII) file
B	Check engineering data for reasonableness
C	Sort an engineering file chronologically
D	Append two sorted engineering files
E	Summarize an engineering file and present results
F	Create table of day, numbers, and corresponding dates
Q	QUIT****

PRESS DESIRED KEY

building. There is no limit, however, to the number of files used for that building while in the routine. To initialize a new building, simply exit to the main menu and reenter the summary routine.

2.3 DISK DRIVE SPECIFICATION

The first prompt the user will see after executing TRANZ will ask for the disk drive where the files are to be located. The user has three choices:

1. For a Single Drive System: Enter 'A:'. Remove the program disk and replace with a disk containing the files. If the program disk is removed, the computer will ask for a system disk if a runtime error occurs or if the user terminates the program. This should cause little problem.
2. For a Double Drive System: Have the program disk in drive 'A:' and the files in drive 'B:', since the computer will look for a system disk in drive 'A:' when the program terminates. The file disk may be

changed during execution as long as the red 'busy' light for the drive is not lit. The new disk must have all the files necessary for the given operation.

3. For a Hard Disk: The program TRANZ.EXE (with BASRUN.EXE) and/or the building and data files may also be stored on a hard disk, usually specified as 'D:'.

2.4 BUILDING AND FILE NAMES

As already noted, each routine prompts for a building name. This name is used with the '.DAT' and '.PAR' extensions to locate these two files (i.e., generally 'building.DAT' and 'building.PAR'). These two files contain information about channel status, description, calibration factors, units, and maximum values; thus they are crucial to proper program operation. These two files are created by another BASIC program for the IBM PC®. This program, called PROGRAM2.BAS (and the compiled version, DIAL.EXE), was also developed by PNL for ELCAP. Example '.DAT' and '.PAR' files are shown in Tables A.1 and A.2 of Appendix A.

Several problems may occur with incorrect building name inputs. Note that the default record length is nine. If the screen shows 'RECORD LENGTH 9' after prompting for a building name, an incorrect or missing building name was probably entered. Reinitialize the program and start again.

Spelling is important. If a misspelled building name is entered, a blank '.DAT' file and '.PAR' file will be created under that name. If file names are misspelled, a disk error will result in most cases, and execution will be stopped. A typical error message is 'File not found at address XXXX:YYYY', where XXXX and YYYY are four digit hexadecimal numbers.

The usual form for raw data file names is 'building.ddd', where 'building' is the building name and 'ddd' is the date of collection (e.g., for July 2, ddd = '702'). A convenient way to label ASCII engineering files is 'buildENGR.ddd', where 'build' is the first four letters of the building

name. Numbers can be used instead of names; e.g., use only the numbers when naming engineering files (7777ENGR.ddd).

If the file name to be created is the same as a file that already exists on the file disk, two things can happen. If that file is already open (i.e., being used), the following runtime error will occur: 'File already open at address XXXX:YYYY'. Program execution will be terminated. If the file is not open, the engineering file being created will be appended (in the same form as if it were given a new name) to the end of the existing file, whatever it may be. This cannot be done as a means of appending without using the append routine because time overlap is not removed, and index numbers are not changed. If the engineering file is appended to a compressed data file, both the compressed data and the engineering data are rendered useless and cannot be accessed. With the sort routine, the user has the option to name the sorted file the same name as the unsorted file, but the unsorted file will be deleted. One should not, however, use the name of any other existing file.

2.5 INKEYS AND USER INPUTS

The pauses in the program suspend operation until the user enters a character from the keyboard. The message 'PRESS ANY KEY TO CONTINUE' is displayed on the screen. Nearly any key will work (except 'shift', 'control', etc.), but do not use the F1 through F10 function keys (left of the main keyboard) because they enter more than one character into the keyboard buffer. Thus, pauses after that point may be skipped because the keyboard buffer already contains a character, and the program does not look for another character from the keyboard. For this reason, do not type into the keyboard buffer during program execution.

2.6 EXAMINING FILES

If questionable program results occur during program execution, look at the current files using the DOS 'TYPE' command after a DOS-system prompt. To obtain a DOS-system prompt the user must first terminate operation of the program. The format for the command is as follows: 'TYPE device:filename.extension'. The 'device' is the drive letter where the file

can be found. Punctuation is important. After the command is entered, the file will be printed on the screen. To temporarily stop the type command, press the 'Num Lock' key while holding the 'control' key down (i.e., control-Num Lock). To continue, press any other key. To terminate the type command, simply use 'control-Break'.

To get a hard copy of the file while using the DOS 'TYPE' command, press control-PrtSc (i.e., Print Screen) before entering the command and after the command is complete. With this method, the user can also produce a hard copy of small sections of the data by entering the control-PrtSc to start printing at a given point and again to stop. For any printer to function properly, the computer must be initialized correctly according to the printer specifications, and the printer must be 'on line'. (If correct initialization is not known, check the printer manual.) When printing, the computer must be correctly initialized. If a hard copy of the whole file is desired, use the DOS 'COPY' command. In this case, the file is not shown on the screen. The format for this command is as follows: 'COPY device:filename.extension LPT1:'. The command sends the file directly to the line printer.

3.0 ROUTINE TO CONVERT RAW DATA TO ASCII ENGINEERING UNITS

When raw data are obtained from the data loggers, they are stored in a file as a succession of one-byte characters. By typing the file on the screen, the user will see what looks like random characters and will probably hear several beeps from the computer. (These beeps are due to control characters in the file.) TRANZ can take the nearly incomprehensible, compressed data and properly convert it to a formatted ASCII engineering unit file only when the correct '.DAT' and '.PAR' files are used. If the wrong files are used, or if one is not current, the data conversion will not work correctly, and the resulting file will be of little use. An incorrect '.PAR' file yields an incorrect record length, and numbers will be read into the wrong category. An incorrect '.DAT' file will give improper calibration of the data. This problem can be difficult to detect because the file will look organized, and yet much or all of the data may be incorrect.

The program allows for data with any integration period to be converted, primarily to accommodate both initial 5-min integration and standard hourly integration. The integration period is determined from the '.PAR' file; again the wrong file could result in incorrect data. If the integration period is wrong, then all data will be scaled incorrectly, though relative magnitudes will be unchanged.

The routine begins by prompting the user for the building name to find the correct '.DAT' and '.PAR' files. The actual prompt is

BUILDING?

After the name is entered, the program returns the record length and prompts for the compressed file name with

COMPRESSED FORMAT FILE NAME?

Once this file name is entered, the number of channels is returned by TRANZ and the ASCII engineering file name is asked for with the prompt

ASCII OUTPUT DATA FILE NAME?

The program then proceeds to convert the compressed file to engineering units and returns to the main menu.

Once the converted file is created, either examine the file from outside the program or use the check routine to ensure that the data are reasonable. The check routine will help spot the 'out of range' values, overflow errors, and problems with the '.DAT' file. If an extreme number of errors are noted on several channels, exit the program and examine the actual file.

4.0 ROUTINE TO CHECK DATA REASONABLENESS

The primary function of this routine is to check the data for values that are 'out of range'. An 'out of range' value is one that is less than zero or greater than the maximum value taken from the '.DAT' file. If the temperature data are in degrees Celsius, an error may occur because of negative data, since below freezing temperatures are common. Because the good data range is flexible, two other errors are noted by the check routine: printing overflows and possible bad '.DAT' file values.

Often during the conversion process, data that were originally erroneous in the logger will convert to numbers that will not fit in the 5-digit field designated for them. In this case, an overflow marker (i.e., the % sign) is printed with the oversized number. Unfortunately, the whole data field becomes offset. Thus, it is necessary to check for and to keep track of overflows as the file is being read in order to keep the data from becoming jumbled.

In addition, some of the converted data may look incorrect because of an incomplete or incorrect '.DAT' file, even though, at least in the raw data file, the data are correct. This is because the maximum value for each channel is obtained using the maximum count value (usually 255) and two calibration factors in the '.DAT' file. The maximum value is calculated from the equation

$$ZMAX(Q) = VAL(CL1\$) * VAL(MV\$) + VAL(CL2\$) \quad (1)$$

where ZMAX(Q) = maximum value for channel Q
VAL(CL1$) = calibration constant number one
VAL(MV$) = maximum number of counts (from '.DAT' file)
VAL(CL2$) = calibration constant number two.

If any of these values is incorrect, the range by which values are checked will be incorrect. For this reason, maximum values that are obviously incorrect (i.e., ≤ 0) are noted.

This routine also begins by prompting for the building name with

BUILDING?

Then, after returning the record length, the program will continue by prompting for the name of the file to be checked with

NAME OF ENGINEERING FILE TO BE CHECKED?

Once the file name has been entered, the program will prompt for the reasonable range for the temperature data with

MINIMUM VALUE FOR TEMPERATURE DATA?

MAXIMUM VALUE FOR TEMPERATURE DATA?

If the maximum value is greater than the minimum value, if the values are equal, or if no range is entered, the default values are set to -50° for the minimum and 60° for the maximum (Celsius temperature range). This prompt allows a switch to kelvin if desired. After this information has been entered, TRANZ asks whether or not a hard copy of the check results is desired with

DO YOU WANT A HARD COPY OF THE RESULTS (Y or N)?

The program then proceeds to check the given data.

The output begins by giving the time period that the data encompasses. Then the error summary output is presented as follows: The kind of error being summarized is printed first. Then 'out of range' values, overflow errors, and '.DAT' file warnings are summarized, respectively. For each channel with an 'out of range' value or an overflow, the channel description and the number of errors for that channel are displayed with the total number of 'out of range' values or overflows. For channels with a maximum value (from the '.DAT' file) ≤ 0, the channel description is noted, and the total number of warnings is stated. If a channel description of '** NO LABEL **' is given, the program probably tried to read nonexistent values at the end of each record, and the message should be of no consequence. If other warnings occur, examine the '.DAT' file or check the parameters with PROGRAM2.BAS or DIAL.EXE. A hard copy of the error summary is optional (an example error summary is shown in Table A.7).

5.0 ROUTINE TO SORT A FILE CHRONOLOGICALLY

Data dumped from the logger are sequential, but the real time order of the records may not be perfectly chronological because of the logger memory configuration. Once data enters memory, it does not move; but when memory is full, the earliest data in memory are replaced with new data. Therefore, even though the data are in order, the earliest time may not be at the beginning of memory block. In most cases, the earliest time is somewhere in the middle of the file. From that point to the end of the file will be chronological, and from the beginning of the file to that point will be chronological. This routine separates the two sections and rewrites them in order.

First the program searches for the earliest time by reading through the file. When determined, the record number for the earliest time is noted. Then, beginning at that record, the file is rewritten in a new file until the end of the file is reached. Subsequently, the beginning portion of the file is rewritten in the same new file after the earlier data. As the new file is created, the records are renumbered. If any power outages are detected (i.e., a day number <100), then those records are noted and are given new 'day numbers' and 'seconds since midnight' data in the sorted file. If all the data are for a power outage, that data are sorted; however, the days and times are not changed because there is no reference from which the values can be obtained.

The routine begins with the standard prompt for the building name:

BUILDING?

After returning the record length, the program then gives the following two prompts in succession:

NAME OF ENGINEERING FILE TO BE SORTED?
ENTER NAME FOR NEW SORTED FILE:
(SAME AS ABOVE FOR WRITE-OVER)?

Once the file names have been entered, TRANZ begins the sorting process. Messages noting what the program is doing are displayed along with the number of power outages. If the file is already sorted, the program gives the user

the option to resave the file with the corrected 'day numbers' and 'seconds since midnight' (for files with power outages) with the following prompt:

RESAVE WITH POWER OUTAGE DAYS/TIMES RESET (Y or N)?

The routine will resave the file even if no power outages were detected during the sorting procedure, if a 'Y' is entered.

If the sorted file is to be given the same name as the unsorted file, the sorted file is first saved under a temporary name. When complete, the unsorted file is deleted, and the sorted file is given its name. Thus, the disk must still have enough space to hold the temporary file before the sort begins.

6.0 ROUTINE FOR APPENDING FILES

This routine will join two files, creating a third file and eliminating any duplicate data. In this way, several files may be appended, two at a time. The program, however, can only handle up to 999 records per file. In this routine, TRANZ gives the following three prompts for file names before beginning the append:

NAME OF FIRST FILE?
NAME OF FILE TO APPEND?
NAME FOR NEWLY APPENDED OUTPUT FILE?

The routine first determines the earliest and latest time for each of the two files to be appended. To be done correctly, both files must already be sorted. The order in which the first two files are entered has no bearing on the appending process. Then the configuration of the two files is determined. If the files overlap (Figure 2a), then the program proceeds to append. If one file is included in the other (Figure 2b), the message

TIME PERIOD FOR file1 INCLUDES ALL OF file2

is given, and the program returns to the main menu. If the files include the same time period (Figure 2c), the message

FILES file1 AND file2 INCLUDE THE SAME TIME PERIOD

is displayed, and the program returns to the main menu. In these two cases the files are not appended. If the files do not overlap (Figure 2d), the message

FILES file1 AND file2 INCLUDE TOTALLY DIFFERENT TIME PERIODS
FILES MAY BE DISJOINT

is given, and the user has the option to continue the append or to abort.

During the actual append process, the file with the earliest time (e.g., file A) is simply rewritten under the append file name. Then the file with the latest time (e.g., file B) is rewritten at the end of the new appended file, beginning with the first record in file B after the last time in file A. The records from file B are renumbered to obtain unduplicated sequential indexes.

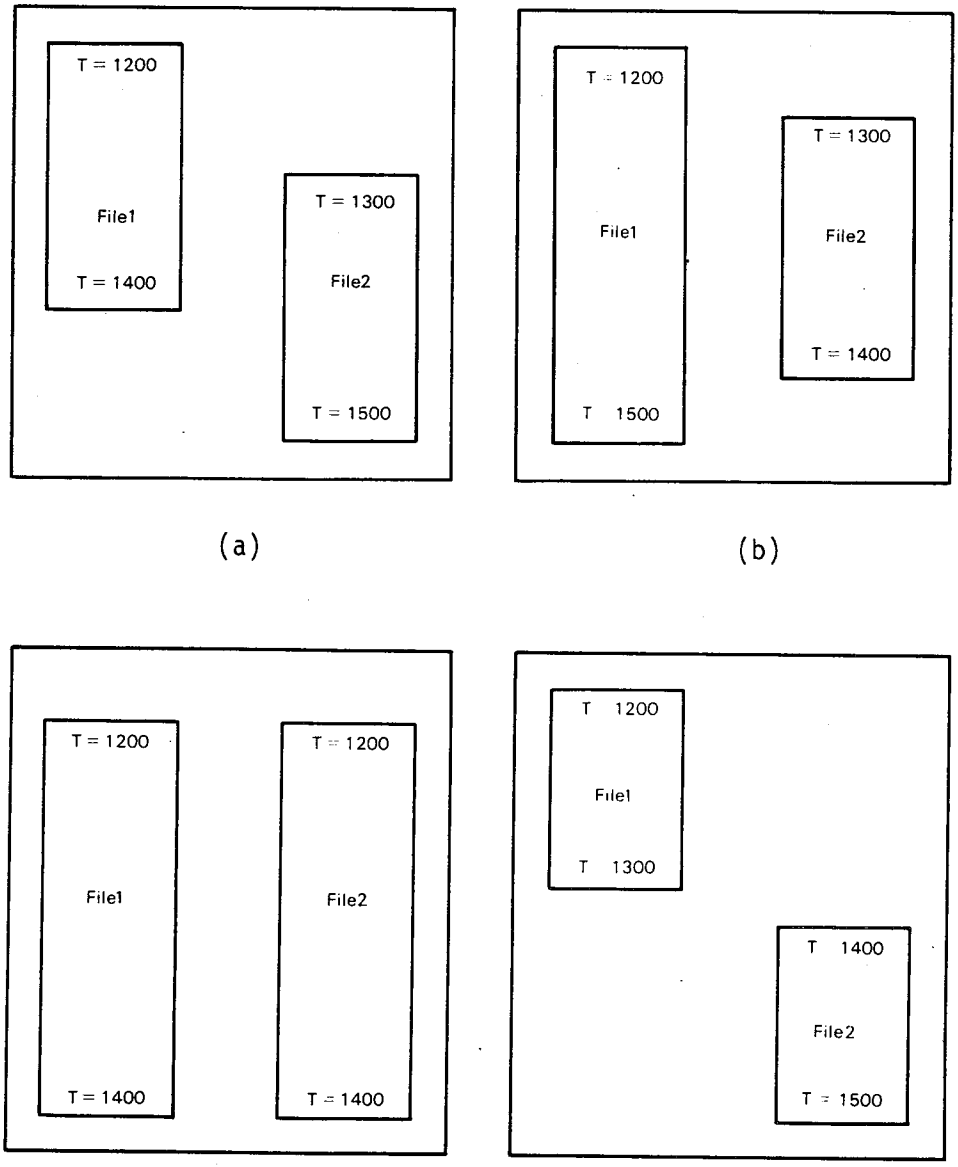

FIGURE 2. Configurations for Files to be Appended
(a) Overlapping Files
(b) One File Included Within Another File
(c) Files Include Same Time Period
(d) Files Do Not Overlap

7.0 ROUTINE TO SUMMARIZE DATA

This routine is divided into two major sections: tabular summaries and graphical summaries. Though the menu (Table 2) has three execution options (A, B, and C), both plotting routines use the same section of code, and a flag determines which of the plots is created. After choosing the summary routine, TRANZ prompts for the building name with

BUILDING?

When that is entered, the record length is returned and the program prompts with

DO YOU WANT A HARD COPY OF 'ON CHANNEL' DESCRIPTIONS (Y/N)?

When the question is answered, the program proceeds to display the 'ON CHANNEL' descriptions in groups of ten. As the program states, the user can return to the 'ON CHANNEL' descriptions at nearly any pause in the summary routine by entering a 'P'. TRANZ then displays the following summary menu:

TABLE 2. Example of Summary Menu

Key	Choice
A	Tabular Summaries (Max, Min, Avg, Std Dev, Total)
B	Scatter Plot of Usage Data
C	Point-to-Point Plot of Usage Data
P	Display 'On Channel' Descriptions
Q	QUIT **** (Return to Main Menu)
	PRESS DESIRED KEY

7.1 TABULAR SUMMARIES

When the tabular summary (choice 'A') is chosen, TRANZ first prompts for the file name with

NAME OF ENGINEERING FILE TO BE SUMMARIZED
(DO NOT ENTER 'P' FOR DESCRIPTIONS)?

All files to be summarized must be sorted, so the routine will work properly. If 'P' is entered at this point, it will be taken as the file name. Next, the program will ask

DO YOU WANT A HARD COPY OF THE TABLE (Y or N)?

This input determines whether or not the resulting summary table is sent to the printer. After this information is received, TRANZ prompts with

NUMBER OF CHANNELS TO BE SUMMARIZED AND TABULATED (1-ALL)?

By entering 'ALL', the program will automatically summarize every channel. The user may also choose a specific number of channels not to exceed the total number of 'ON CHANNELS'. If something besides 'ALL' or a number is entered, a zero is assumed for the number of channels to be summarized. Even though the program will search the file, no actual summary will be generated. If a number other than 'ALL' is entered, the program will prompt with

ENTER 'ON CHANNEL' NUMBERS AS GIVEN IN THE TABLE
(OR 'P' FOR DESCRIPTION DISPLAY)?

Choosing specific 'ON CHANNELS' may be desirable when making comparisons because the channels can be summarized in any order. If an 'ON CHANNEL' number is chosen that does not exist, the program assigns '** NO LABEL **' to that 'ON CHANNEL' description. If an incorrect 'ON CHANNEL' number is accidentally entered, the process can be reinitialized by entering 'P'.

The tabular summary yields five types of information for each 'ON CHANNEL':

1. minimum value for the channel
2. maximum value for the channel
3. average value for the channel
4. standard deviation for all channel values
5. total of all channel values.

As the program reads through each channel, it keeps a running tally of the sum of the channel values, the sum of the square of the channel values, and the

number of values. The program simultaneously keeps track of the minimum and maximum for the channel. The standard deviation is calculated using the following equation:

$$SDEV = SQR((X_i^2 - n * X_{avg}^2)/(n - 1)) \quad (2)$$

where SDEV = the standard deviation for the channel i

SQR = the square root function

X_i = an individual value for the channel i

n = the number of values for the channel i

X_{avg} = the average value for the channel i.

If the argument of the square root function is less than zero (negative), then the argument is set to zero. When an overflow is encountered, the maximum, minimum, and average will be 99999. An example of channel summary output can be found in Table A.9 of Appendix A.

7.2 GRAPHICAL SUMMARIES

When trying to understand trends in the data, the graphical summaries are the most useful tool. Files to be plotted must be previously sorted in order for the graphical routine to work correctly.

When entering the routine, the program asks if the user is using a graphics capable screen with the prompt

DO YOU HAVE A GRAPHICS CAPABLE SCREEN (Y/N)?

This option allows those users who have a plotter but no graphics screen to generate hard copy plots. If a graphics screen is not being used and a 'Y' is entered, the error message 'Device unavailable at XXXX:YYYY' will appear, and program execution will be terminated. The next prompt that appears in the program is

NAME OF SORTED FILE TO BE PLOTTED?

When the file name is received, TRANZ asks for the 'ON CHANNEL' that is to be plotted with

ON CHANNEL NUMBER FOR PLOT TO BE CREATED?

If the number is not known, the user can go to the 'ON CHANNEL' descriptions by entering 'P'. If an on 'ON CHANNEL' number is given that does not exist, the program tells the user that no data exists or that all quantities have values of zero. Next, TRANZ prompts for the time period that the plot will include according to the following format:

ELCAP DAY 1 = JANUARY 1, 1981
ELCAP DAY NUMBER AT WHICH PLOT WILL BEGIN ('ALL' FOR FILE)?
TIME (IN SECONDS) FOR THE ABOVE DAY?
ELCAP DAY NUMBER AT WHICH PLOT WILL END?
TIME (IN SECONDS) FOR THE ABOVE DAY?

Entering 'ALL' after the first prompt will skip the other prompts and will automatically plot all the data in the file. If nothing is entered for a prompt, then a zero value is assigned. If any other word besides 'ALL' is entered for the first prompt, zero is assigned. Obviously, if the time period that the data encompasses is not known, start with the 'ALL' option. Once the entire file is plotted, the user can focus on smaller portions of the timeline.

Once the plot has been generated on the screen, it remains until the user enters a character from the keyboard. When the character is received, TRANZ prompts with

DO YOU WANT A HARD COPY OF THE PLOT (Y/N)?

If an 'N' is entered, the program branches around the hard copy plotting section and asks whether another plot from the same file is desired. If a 'Y' is entered, the program prompts with

DO YOU HAVE AN HP PLOTTER? (A = 7475A, B = 7470A, N = NONE)

The program was designed only for the HP 7470A and the HP 7475A plotters. It may, however, be possible to use the program with other plotters based on the HP-GL plotter language used in this program. Some experimentation will probably be required. If an 'N' is entered, the program jumps to the end of the routine and asks whether another plot from the same file is desired. If either 'A' or 'B' is entered, the program will continue with the prompts

LOAD PAPER FOR PLOT AND PREPARE PLOTTER
PRESS ANY KEY TO CONTINUE

This pause allows the user to get the paper loaded correctly and to make certain that the plotter is properly connected to the IBM PC®. This pause is especially useful for computers with only one communications port, where the printer may have to be disconnected in order to connect the plotter. When the program continues, it prompts for the communications port with

COMMUNICATIONS PORT NUMBER (COM1, COM2)?

If anything other than the two specified inputs (i.e., COM1, COM2) is entered, the error message 'Device unavailable at XXXX:YYYY' is displayed, and program execution is terminated. This option allows for computers that may have more than one communications port to choose from, eliminating the need to switch cables in the back of the machine.

If the plotter specified is the HP 7475A, then the program will prompt for the paper size with

PAPER SIZE FOR PLOT? (1 = 11 X 17, 2 = 8.5 X 11)?

Though '1' and '2' are specified as inputs, any whole number ≤ 1 will set the 11- x 17-in. paper size, and any other whole number will set the 8.5- x 11-in. paper size.

After the plot is finished, TRANZ displays the message

WANT ANOTHER ORIGINAL OF THE SAME PLOT (Y/N)?

If a 'Y' is entered, then the same plot will be generated again, beginning with the pause for loading the paper. The computer may show this prompt before the plotter is actually finished plotting, so the user should wait until the plot is complete before proceeding. If an 'N' is entered, then the program prompts with

PLOT ANOTHER CHANNEL FROM THE SAME FILE (Y or N)?

If a 'Y' is entered, the routine starts again at the prompt for the 'ON CHANNEL' to be plotted. The second plot that is generated will be the same

type as the first (i.e., scatter or point-to-point). If an 'N' is entered, then the program returns to the summary menu.

Using the user-defined limits, the program gathers the data. Then after determining the maximum and minimum values on both axes, the program sets the configuration and increments for the axes. The horizontal time axis will always round to the next lowest and the next highest day when setting the axis length, although only data for the chosen time period is plotted. The vertical axis always ranges from the minimum value (or zero, whichever is lower) to the maximum value. Negative data will be plotted correctly. If a power outage gap is found in the data file, a line will connect the bounding data points on a point-to-point plot. This fact may be helpful in explaining strange data characteristics. The label and units for the vertical axis are given above the plot (to the left of the plot on the hard copies). An example of a scatter plot and a point-to-point plot are shown in Figures A.1 and A.2, respectively, of Appendix A.

8.0 ROUTINE TO CREATE A TABLE OF DAY NUMBERS AND DATES

This routine will print a one-to-three column table of day numbers and their corresponding date. The day of the week is also included. When the routine is specified, the following prompts appear on the screen:

ELCAP DAY NUMBER TO START THE TABLE?
<JANUARY 1, 1981=1> <JANUARY 1, 1982=366> <JANUARY 1, 1983=731>
<JANUARY 1, 1984=1096> <JANUARY 1, 1985=1462> <JANUARY 1, 1986=1827>
ELCAP DAY NUMBER TO END THE TABLE?

The user defines the limits of the table using day numbers. Any time period since January 1, 1981, can be tabulated. Every day number for the period is printed for the user's ease in determining the corresponding date. Day numbers and dates are printed both on the screen and on the printer. An example of the output for this routine is given in Table A.10 in Appendix A.

APPENDIX A

SAMPLE FILES AND PROGRAM OUTPUT

APPENDIX A

SAMPLE FILES AND PROGRAM OUTPUT

Several files and printed program outputs have been included in this appendix to show the file types under TRANZ. The building name has been changed to 'BUILDING' to ensure anonimity.

The '.DAT' file (Table A.1) is made up of 112 records (corresponding to the maximum number of channels: 48 digital and 64 analog), each with a length of 54 characters. The field is divided into six sections: the title, calibration factor one (multiplier), calibration factor two (offset), units, accuracy (*), and maximum counts. The file is used for identification of channels and data conversion. This file [along with the '.PAR file (see Table A.2)] is created using the program PROGRAM.BAS or DIAL.EXE. These programs were developed for the End-Use Load and Conservation Assessment Program (ELCAP) by Pacific Northwest Laboratory (PNL).

The '.PAR' file is composed of 239 records, each with a length of four characters. The various records contain information on such topics as channel status and integration period.

The first data file (Table A.3) is 'BUILENGR.831'. This file shows the form of the data just after it has been converted, although the file has been shortened for easy display. The 'ON CHANNELS' begin in column seven. Column one is the index, column two is the check sum, column three is the record length, and column six is the wind direction. Columns four and five are particularly important to this program. Column four is the day number, and column five is the number of seconds after midnight. These six columns should contain the same type of information in all files. One can determine the on-channel information by correlating the data with the sequential entries in the '.DAT' file. Note that the second half of this file is power outage data. Also note that the reference column contains overflow data.

The second file (Table A.4) is "BUILSORT.831'. This file is the sorted version of the previous file. Note that none of the data has been changed, but

TABLE A.1. Example of a '.DAT' File Used for Identification of Channels and Data Conversion

```
BUILDING.DAT
 0
WIND                 .75                  m/S  * 225
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
TEMP OUTSIDE         .39       -29.5      DEG C* 255
TEMP CRAWL SPACE     .39       -29.5      DEG C* 255
PYRANOMETER          4.9                  W/M^2* 255
REL HUMIDITY INSIDE  .402                 %    * 255
WIND DIRECTION       1.4116               DEG    255
TEMP BEDROOM #2      .39       -29.5      DEG C* 255
 0
REFERENCE            1.                   COUNTS 255
 0
 0
 0
 0
CO MONITOR                                       255
AEROSOL MONITOR                                  255
 0
 0
TOTAL ELECTRIC PH1   78.8                 WATTS* 255
HOT TUB              46.8                 WATTS* 255
RANGE                18.2                 WATTS* 255
KITCHEN (REFIG)      9.44                 WATTS* 255
MICROWAVE            9.47                 WATTS* 255
LITE                 7.86                 WATTS* 255
LITE                 7.92                 WATTS* 255
DISHWASHER           7.85                 WATTS* 255
TOTAL ELECTRIC PH2   77.7                 WATTS* 255
FURNACE              25.9                 WATTS* 255
HOT WATER            18.5                 WATTS* 255
WASHER               9.48                 WATTS* 255
KITCHEN              9.48                 WATTS* 255
LITE                 7.92                 WATTS* 255
LITE                 7.84                 WATTS* 255
GARB DISPOSAL        7.90                 WATTS* 255
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
```

TABLE A.2. Example of '.PAR' File Containing Information on Channel Status and Integration Periods

BUILDING.PAR				
0	0	12	0	0
0	0	12	0	0
0	0	12	0	0
0	0	0	0	0
0	0	0	0	0
14	0	0	0	0
16	0	0	0	0
64	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
245	0	0	0	0
12	0	0	0	0
255	0	0	0	
255	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
64	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
0	0	0	0	
245	12	0	0	
0	12	0	0	
127	12	0	0	
127	12	0	0	
0	4	0	0	
0	12	0	0	
0	0	0	0	
0	12	0	0	
0	0	0	0	
11	0	0	0	
0	0	0	0	
0	0	0	0	
0	4	0	0	
0	4	0	0	
0	0	0	0	
0	0	0	0	
0	4	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	
0	4	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	
0	12	0	0	

TABLE A.3. 'BUILENGR.831' (First Data File)

```
A>TYPE BUILENGR.831
  1 1441   38 1338 33395  144    2   13   16    0   43   26 %917504    0    0    0    0    0  141    0  152    0    0    0    0  358    0    0    0    0    9
  2 1462   38 1338 36995  164    1   13   16    0   43   26 %917504    0    0    0    0    0  226    0    0    0    0    0    0    0    0    0    0    0   27
  3 1421   38 1338 40595  106    1   13   16    0   43   25 %917504    0    0    0    0    0  118    0    0    0    0    0    0    0    0    0    0    0    9
  4 1499   38 1338 44195  135    2   12   16    0   43   25 %917504    0    0    0    0    0  107    0  161    0    0    0    0    0    0    0    0    0   27
  5 1517   38 1338 47795  138    1   12   16    0   43   25 %917504    0    0    0    0    0  118    0   18    0    0    0    0   84    0    0    0    0    9
  6 1572   38 1338 51395  121    1   12   16    0   43   25 %917504    0    0    0    0    0  129    0   27    0    0    0    0  674    0    0   72   18   27
  7 1696   38 1338 54995  144    0   12   16   33   44   25 %917504    0    0    9    0    0   97    0  224    0    0    0    0 1726    0    0   18   54    9
  8 1669   38 1338 58595  167    1   14   16  173   44   25 %917504    0    0    0    0    0  129    0    0    0    0    0    0    0    0    0    0    0   27
  9 1684   38 1338 62195  124    2   16   16  312   44   25 %917504    0    0    0    0    0  107    0    0    0    0    0    0    0    0    0    9    0    9
 10 1279   38 1338 65795  128    2   17   16  435   44   25 %917504    0    0    0    0    0  107   97  134    9    0    0    0  379    0    0    9    0   27
 11 1412   38    0  3600  223    1   21   16  496   44   25 %917504    0    0    0    0    0  139    0    0   27  723    0    0  316    0    0    9    0   27
 12 1405   38    0  7200  217    2   18   16  117   44   25 %917504    0    0    0    0    0  107    0    0    0  509    0    0 1221    0    0    0    0    9
 13 1096   38    0 10800   13    1   15   16  128   44   25 %917504    0    0    0    0    0   97    0  152    0    9    0    0    0    0    0    9    0    9
 14 1224   38    0 14400   13    0   17   16  117   45   26 %917504    0    0    0    0    0  247    0    0    0    0    0    0 1873    0    0   36   45   27
 15 1372   38    0 18000  229    0   16   16   50   45   26 %917504    0    0    0    0    0  118    0   36   36    0    0    0  958    0    0    9    0    9
 16 1292   38    0 21600  131    1   15   16   22   44   25 %917504    0    0    0    0    0  118    0  197    0    0    0    0    0    0    0    0    0   27
 17 1318   38    0 25200  150    2   13   16   17   45   25 %917504    0    0    0    0    0  129    0    0    0    0    0    0    0    0    0   18    0    9
 18 1414   38    0 28800  151    1   12   16    6   45   25 %917504    0    0    0    0    0  107    0  241    0    0    0    0  548    0    0  171    0   27
 19 1450   38    0 32400  214    1   11   15    0   45   25 %917504    0    0    0    0    0   97    0    9    0    0    0    0  337    0    0   27    0   27
 20 1420   38    0 36000  129    0   10   16    0   45   25 %917504    0    0    0    0    0  140    0  170  144    0    0    0    0    0    0  117    0    9
```

TABLE A.4. 'BUILSORT.831' (Second Data File)

```
A)TYPE BUILSORT.831
  1  1441   38  1338 33395  146   2  13  16    0  43  26 %917504   0   0   0   0   0  161   0  152   0   0   0   0  358   0   0   0   0   9
  2  1462   38  1338 36995  169   1  13  16    0  43  26 %917504   0   0   0   0   0  226   0    0   0   0   0   0    0   0   0   0   0  27
  3  1421   38  1338 40595  106   1  13  16    0  43  25 %917504   0   0   0   0   0  110   0    0   0   0   0   0    0   0   0   0   0   9
  4  1499   38  1338 44195  135   2  12  16    0  43  25 %917504   0   0   0   0   0  107   0  161   0   0   0   0    0   0   0   0   0  27
  5  1517   38  1338 47795  138   1  12  16    0  43  25 %917504   0   0   0   0   0  110   0   18   0   0   0   0   84   0   0   0   0   9
  6  1572   38  1338 51395  121   1  12  16    0  43  25 %917504   0   0   0   0   0  129   0   27   0   0   0   0  674   0   0  72  18  27
  7  1696   38  1338 54995  144   0  12  16   33  44  25 %917504   0   0   0   0   0   97   0  224   0   0   0   0 1726   0   0  18  54   9
  8  1669   38  1338 58595  167   1  14  16  173  44  25 %917504   0   0   0   0   0  129   0    0   0   0   0   0    0   0   0   0   0  27
  9  1686   38  1338 62195  124   2  16  16  312  44  25 %917504   0   0   0   0   0  107   0    0   0   0   0   0    0   0   0   9   0   9
 10  1279   38  1338 65795  128   2  17  16  435  44  25 %917504   0   0   0   0   0  107  97  134   9   0   0   0  379   0   0   9   0  27
 11  1412   38  1338 69395  223   1  21  16  496  44  25 %917504   0   0   0   0   0  129   0    0  27  723   0   0  316   0   0   9   0  27
 12  1405   38  1338 72995  247   2  18  16  117  44  25 %917504   0   0   0   0   0  107   0    0   0  509   0   0 1221   0   0   0   0   9
 13  1096   38  1338 76595   13   1  15  16  128  44  25 %917504   0   0   0   0   0   97   0  152   0   9   0   0    0   0   0   9   0   9
 14  1224   38  1338 80195   13   0  17  16  117  45  26 %917504   0   0   0   0   0  247   0    0   0   0   0   0 1873   0   0  36  45  27
 15  1372   38  1338 83795  229   0  16  16   50  45  26 %917504   0   0   0   0   0  118   0   36  36   0   0   0  358   0   0   9   0   9
 16  1292   38  1339   995  131   1  15  16   22  44  25 %917504   0   0   0   0   0  118   0  197   0   0   0   0    0   0   0   0   0  27
 17  1318   38  1339  4595  150   2  13  16   17  45  25 %917504   0   0   0   0   0  129   0    0   0   0   0   0    0   0   0  18   0   9
 18  1414   38  1339  8195  151   1  12  16    6  45  25 %917504   0   0   0   0   0  107   0  241   0   0   0   0  568   0   0 171   0  27
 19  1450   38  1339 11795  214   1  11  16    0  45  25 %917504   0   0   0   0   0   97   0    9   0   0   0   0  337   0   0  27   0  27
 20  1420   38  1339 15395  129   0  18  16    0  45  25 %917504   0   0   0   0   0  140   0  170 144   0   0   0    0   0   0 117   0   9
```

the day numbers and the seconds since midnight for the power outage have been reset. This file was used for subroutines 'B' and 'D'.

The third file (Table A.5) is 'BUILDENGR.830'. This sorted engineering file contains data prior to the file 'BUILSORT.831' and was appended to that file. The appended file 'BUILAPND.831' (Table A.6) contains 35 records. Note that five repetitious records were not included a second time.

A reasonableness check summary for 'BUILSORT.831' is given in Table A.7. The reference has 20 'out of range' values and 20 overflows. A quick check of the '.DAT' file shows that neither the 'CO MONITOR' nor the 'AEROSOL MONITOR' has any calibration factors. Thus, their maximum values were computed to be zero, and they were included in the listed warnings. Also note the example of a '** NO LABEL **' warning. This warning is of no consequence, unless it comes in the middle of the list. If so, the '.PAR' and '.DAT' files should be checked for errors.

The summary routine was run with the appended file 'BUILAPND.831'. A list of 'ON CHANNEL' descriptions as the program would print them is given in Table A.8. These descriptions can be checked in the '.DAT' file. Option 'A' of the summary menu produced the tabular summary found in Table A.9. A manual check will verify the accuracy of the output. Option 'B' was used with the same file for the outside temperature data ('ON CHANNEL' #2), and the scatter plot found in Figure A.1 was generated. A point-to-point plot (Figure A.2) was produced for the refrigerator ('ON CHANNEL' #13) using option 'C' of the summary menu.

Table A.10 gives an example of the day number tables produced by the main routine 'F'.

TABLE A.5. 'BUILENGR.830' (Third Data File)

```
A)TYPE BUILENGR.830
  1  1271   38  1337 65795  124   3   23   16   591   39   24 %917504   0   0   0   0   0  129   0   45   0   0   0   0   41   0   0   0   0  27
  2  1342   38  1337 69395  150   3   26   17   680   40   24 %917504   0   0   0   0   0   97  11    0   9   0   0   0    0   0   0   9   0   9
  3  1397   38  1337 72995  101   3   27   17   543   40   24 %917504   0   0   0   0   0  129  11    0  54   0   0   0    0   0   0   9   0  27
  4  1401   38  1337 76595  147   2   29   18   552   40   25 %917504   0   0   0   0  21  118   0  134   0   0   0   0    0   0   0   9   0   9
  5  1483   38  1337 80195  141   2   29   18   613   41   25 %917504   0   0   0   0 331  118   0  233   0   0   0   0  358   0   0   9   0   9
  6  1527   38  1337 83795  220   2   29   18   329   41   26 %917504   0   0   0   0 166  247   0   18   0   0   0   0  358   0   0  27   0  27
  7  1546   38  1338   995  228   3   28   18   184   40   26 %917504   0   0   0   0   0  193   0    0   0   0   0   0    0   0   0   0   0   9
  8  1416   38  1338  4595  114   3   28   17     0   40   26 %917504   0   0   0   0   0  150   0    0  54   0   0   0    0   0   0   9   0  27
  9  1351   38  1338  8195  174   1   16   16     6   42   26 %917504   0   0   0   0  21  140   0    0 189   0   0   0 1958   0   0  18  36   9
 10  1307   38  1338 11795  120   1   16   16     6   43   26 %917504   0   0   0   0 352  150  11  152 342   0   0   0  189   0   0 126   0  27
 11  1311   38  1338 15395  135   2   15   16     0   44   26 %917504   0   0   0   0   0  141   0    0 288   0   0   0    0   0   0 216   0   9
 12  1354   38  1338 18995  144   2   15   16     0   43   26 %917504   0   0   0   0   0  107   0  170 153   0   0   0  218   0   0 171   0   9
 13  1344   38  1338 22595  128   1   14   16     0   43   26 %917504   0   0   0   0   0  150   0   54   0   0   0   0  737   0   0   0   0  27
 14  1360   38  1338 26195  142   2   14   16     0   43   26 %917504   0   0   0   0   0  107   0  170   0   0   0   0    0   0   0   0   0   9
 15  1374   38  1338 29795  146   2   13   16     0   43   26 %917504   0   0   0   0   0  107   0    0   0   0   0   0    0   0   0   0   0  27
 16  1441   38  1338 33395  146   2   13   16     0   43   26 %917504   0   0   0   0   0  141   0  152   0   0   0   0  358   0   0   0   0   9
 17  1462   38  1338 36995  164   1   13   16     0   43   26 %917504   0   0   0   0   0  226   0    0   0   0   0   0    0   0   0   0   0  27
 18  1421   38  1338 40595  106   1   13   16     0   43   25 %917504   0   0   0   0   0  118   0    0   0   0   0   0    0   0   0   0   0   9
 19  1499   38  1338 44195  135   2   12   16     0   43   25 %917504   0   0   0   0   0  107   0  141   0   0   0   0    0   0   0   0   0  27
 20  1517   38  1338 47795  138   1   12   16     0   43   25 %917504   0   0   0   0   0  118   0   18   0   0   0   0   84   0   0   0   0   9
```

TABLE A.6. 'BUILAPND.831' (Appended File)

A)TYPE BUILAPND.831

1	1271	38	1337	65795	124	3	23	16	591	39	24	917504	0	0	0	0	0	129	0	45	0	0	0	0	42	0	0	0	0	27
2	1342	38	1337	69395	150	3	26	17	680	40	24	917504	0	0	0	0	0	97	11	0	9	0	0	0	0	0	0	9	0	9
3	1397	38	1337	72995	181	3	27	17	563	40	24	917504	0	0	0	0	0	129	11	0	54	0	0	9	0	0	0	9	0	27
4	1401	38	1337	76595	147	2	29	18	552	40	25	917504	0	0	0	0	21	118	0	136	0	0	0	0	0	0	0	9	0	9
5	1483	38	1337	80195	141	2	29	18	613	41	25	917504	0	0	0	0	331	118	0	233	0	0	0	0	358	0	0	9	0	9
6	1527	38	1337	83795	220	2	29	18	329	41	26	917504	0	0	0	0	146	247	0	18	0	0	0	0	358	0	0	27	0	27
7	1546	38	1338	995	228	3	28	18	184	40	26	917504	0	0	0	0	0	193	0	0	0	0	0	0	0	0	0	0	0	9
8	1414	38	1338	4595	114	3	20	17	0	40	24	917504	0	0	0	0	0	150	0	0	54	0	0	0	0	0	0	9	0	27
9	1351	38	1338	8195	174	1	16	16	6	42	24	917504	0	0	0	0	21	140	0	0	189	0	0	0	1958	0	0	18	36	9
10	1307	38	1338	11795	120	1	16	16	6	43	24	917504	0	0	0	0	352	150	11	152	342	0	0	0	189	0	0	126	0	27
11	1311	38	1338	15395	135	2	15	16	0	44	24	917504	0	0	0	0	0	161	0	0	288	0	0	0	0	0	0	216	0	9
12	1354	38	1338	18995	144	2	15	16	0	43	24	917504	0	0	0	0	0	197	0	170	153	0	0	0	210	0	0	171	0	9
13	1344	38	1338	22595	128	1	14	16	0	43	24	917504	0	0	0	0	0	150	0	54	0	0	0	0	737	0	0	0	0	27
14	1360	38	1338	26195	142	2	14	16	0	43	24	917504	0	0	0	0	0	107	0	170	0	0	0	0	0	0	0	0	0	9
15	1374	38	1338	29795	146	2	13	16	0	43	24	917504	0	0	0	0	0	107	0	0	0	0	0	0	0	0	0	0	0	27
16	1441	38	1338	33395	146	2	13	16	0	43	24	917504	0	0	0	0	0	161	0	152	0	0	0	0	358	0	0	0	0	9
17	1462	38	1338	36995	164	1	13	16	0	43	24	917504	0	0	0	0	0	226	0	0	0	0	0	0	0	0	0	0	0	27
18	1421	38	1338	40595	106	1	13	16	0	43	25	917504	0	0	0	0	0	118	0	0	0	0	0	0	0	0	0	0	0	9
19	1499	38	1338	44195	135	2	12	16	0	43	25	917504	0	0	0	0	0	107	0	161	0	0	0	0	0	0	0	0	0	27
20	1517	38	1338	47795	138	1	12	16	0	43	25	917504	0	0	0	0	0	118	0	18	0	0	0	0	84	0	0	0	0	9
21	1572	38	1338	51395	121	1	12	16	0	43	25	917504	0	0	0	0	0	129	0	27	0	0	0	0	674	0	0	72	18	27
22	1696	38	1338	54995	144	0	12	16	33	44	25	917504	0	0	0	0	0	97	0	224	0	0	0	0	1726	0	0	18	54	9
23	1669	38	1338	58595	167	1	14	16	173	44	25	917504	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0	0	27
24	1686	38	1338	62195	121	2	16	16	312	44	25	917504	0	0	0	0	0	107	0	0	0	0	0	0	0	0	0	9	0	9
25	1279	38	1338	65795	128	2	17	16	435	44	25	917504	0	0	0	0	0	107	97	134	9	0	0	0	379	0	0	9	0	27
26	1412	38	1338	69395	223	1	21	16	496	44	25	917504	0	0	0	0	0	129	0	0	27	723	0	0	316	0	0	9	0	27
27	1405	38	1338	72995	247	2	18	16	117	44	25	917504	0	0	0	0	0	107	0	0	0	589	0	0	1221	0	0	0	0	9
28	1096	38	1338	76595	13	1	15	16	128	44	25	917504	0	0	0	0	0	97	0	152	0	9	0	0	0	0	0	9	0	9
29	1224	38	1338	80195	13	0	17	16	117	45	26	917504	0	0	0	0	0	247	0	0	0	0	0	0	1873	0	0	36	45	27
30	1372	38	1338	83795	229	0	16	16	50	45	26	917504	0	0	0	0	0	118	0	36	36	0	0	0	358	0	0	9	0	9
31	1292	38	1339	995	131	1	15	16	22	44	25	917504	0	0	0	0	0	118	0	197	0	0	0	0	0	0	0	0	0	27
32	1318	38	1339	4595	158	2	13	16	17	45	25	917504	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	18	0	9
33	1414	38	1339	8195	151	1	12	16	6	45	25	917504	0	0	0	0	0	107	0	241	0	0	0	0	568	0	0	171	0	27
34	1450	38	1339	11795	214	1	11	16	0	45	25	917504	0	0	0	0	0	97	0	9	0	0	0	0	337	0	0	27	0	27
35	1420	38	1339	15395	129	0	10	16	0	45	25	917504	0	0	0	0	0	140	0	170	144	0	0	0	0	0	0	117	0	9

TABLE A.7. Reasonableness Check Summary for 'BUILSORT.831'

DATA SPANS THE TIME PERIOD FROM: 8/30/1984 TO 8/31/1984

******* SUMMARY OF ERRORS *******

REFERENCE 20 ERRORS

TOTAL NUMBER OF OUT OF RANGE VALUES IS 20

******* SUMMARY OF OVERFLOWS *******

REFERENCE 20 OVERFLOWS

TOTAL NUMBER OF OVERFLOWS = 20

******* SUMMARY OF WARNINGS *******

MAXIMUM VALUE FOR CO MONITOR	LESS THAN OR EQUAL TO ZERO
MAXIMUM VALUE FOR AEROSOL MONITOR	LESS THAN OR EQUAL TO ZERO
MAXIMUM VALUE FOR ** NO LABEL **	LESS THAN OR EQUAL TO ZERO

TOTAL NUMBER OF WARNINGS IS 3

TABLE A.8. List of 'ON CHANNEL' Descriptions

DESCRIPTIONS FOR °ON CHANNELS' 1 TO 10	
ON CHANNEL # 1	WIND
ON CHANNEL # 2	TEMP OUTSIDE
ON CHANNEL # 3	TEMP CRAWL SPACE
ON CHANNEL # 4	PYRANOMETER
ON CHANNEL # 5	REL HUMIDITY INSIDE
ON CHANNEL # 6	TEMP BEDROOM #2
ON CHANNEL # 7	REFERENCE
ON CHANNEL # 8	CO MONITOR
ON CHANNEL # 9	AEROSOL MONITOR
ON CHANNEL # 10	TOTAL ELECTRIC PH1
DESCRIPTIONS FOR °ON CHANNELS' 11 TO 20	
ON CHANNEL # 11	HOT TUB
ON CHANNEL # 12	RANGE
ON CHANNEL # 13	KITCHEN (REFIG)
ON CHANNEL # 14	MICROWAVE
ON CHANNEL # 15	LITE
ON CHANNEL # 16	LITE
ON CHANNEL # 17	DISHWASHER
ON CHANNEL # 18	TOTAL ELECTRIC PH2
ON CHANNEL # 19	FURNACE
ON CHANNEL # 20	HOT WATER
DESCRIPTIONS FOR °ON CHANNELS' 21 TO 25	
ON CHANNEL # 21	WASHER
ON CHANNEL # 22	KITCHEN
ON CHANNEL # 23	LITE
ON CHANNEL # 24	LITE
ON CHANNEL # 25	GARB DISPOSAL

TABLE A.9. Tabular Summary Produced by Option 'A' of the Summary Menu

ELCAP DAY NUMBER (DAY 1 = JANUARY 1, 1981)-- 1337 TO 1339

DATES INCLUDE: 8/29/1984 TO 8/31/1984

SUMMARIES FOR CHANNELS 1 TO 10 BUILDING

ON CHANNEL	UNITS	MINIMUM	MAXIMUM	AVERAGE	STAND DEV	TOT (W-H)
1 WIND	m/S	0	3	2	0.886	54.0E+00
2 TEMP OUTSIDE	DEG C	10	29	17	5.773	59.6E+01
3 TEMP CRAWL SPACE	DEG C	16	18	16	0.676	57.1E+01
4 PYRANOMETER	W/M^2	0	680	155	224.888	54.3E+02
5 REL HUMIDITY INSIDE	%	39	45	43	1.738	15.0E+02
6 TEMP BEDROOM #2	DEG C	24	26	25	0.631	88.6E+01
7 REFERENCE	COUNT	99999	99999	99999	0.000	35.0E+05
8 CO MONITOR		0	0	0	0.000	0.0E+00
9 AEROSOL MONITOR		0	0	0	0.000	0.0E+00
10 TOTAL ELECTRIC PH1	WATTS	0	0	0	0.000	0.0E+00

SUMMARIES FOR CHANNELS 11 TO 20 BUILDING

ON CHANNEL	UNITS	MINIMUM	MAXIMUM	AVERAGE	STAND DEV	TOT (W-H)
11 HOT TUB	WATTS	0	0	0	0.000	0.0E+00
12 RANGE	WATTS	0	352	25	83.880	89.1E+01
13 KITCHEN (REFIG)	WATTS	97	247	134	39.398	46.9E+02
14 MICROWAVE	WATTS	0	97	4	16.529	13.0E+01
15 LITE	WATTS	0	241	71	85.535	25.0E+02
16 LITE	WATTS	0	342	37	83.865	13.1E+02
17 DISHWASHER	WATTS	0	723	35	147.318	12.4E+02
18 TOTAL ELECTRIC PH2	WATTS	0	0	0	0.000	0.0E+00
19 FURNACE	WATTS	0	0	0	0.000	0.0E+00
20 HOT WATER	WATTS	0	1958	336	546.332	11.7E+03

SUMMARIES FOR CHANNELS 1 TO 5 BUILDING

ON CHANNEL	UNITS	MINIMUM	MAXIMUM	AVERAGE	STAND DEV	TOT (W-H)
21 WASHER	WATTS	0	0	0	0.000	0.0E+00
22 KITCHEN	WATTS	0	0	0	0.000	0.0E+00
23 LITE	WATTS	0	216	32	56.727	11.1E+02
24 LITE	WATTS	0	54	4	13.164	15.3E+01
25 GARB DISPOSAL	WATTS	9	27	18	9.128	62.1E+01

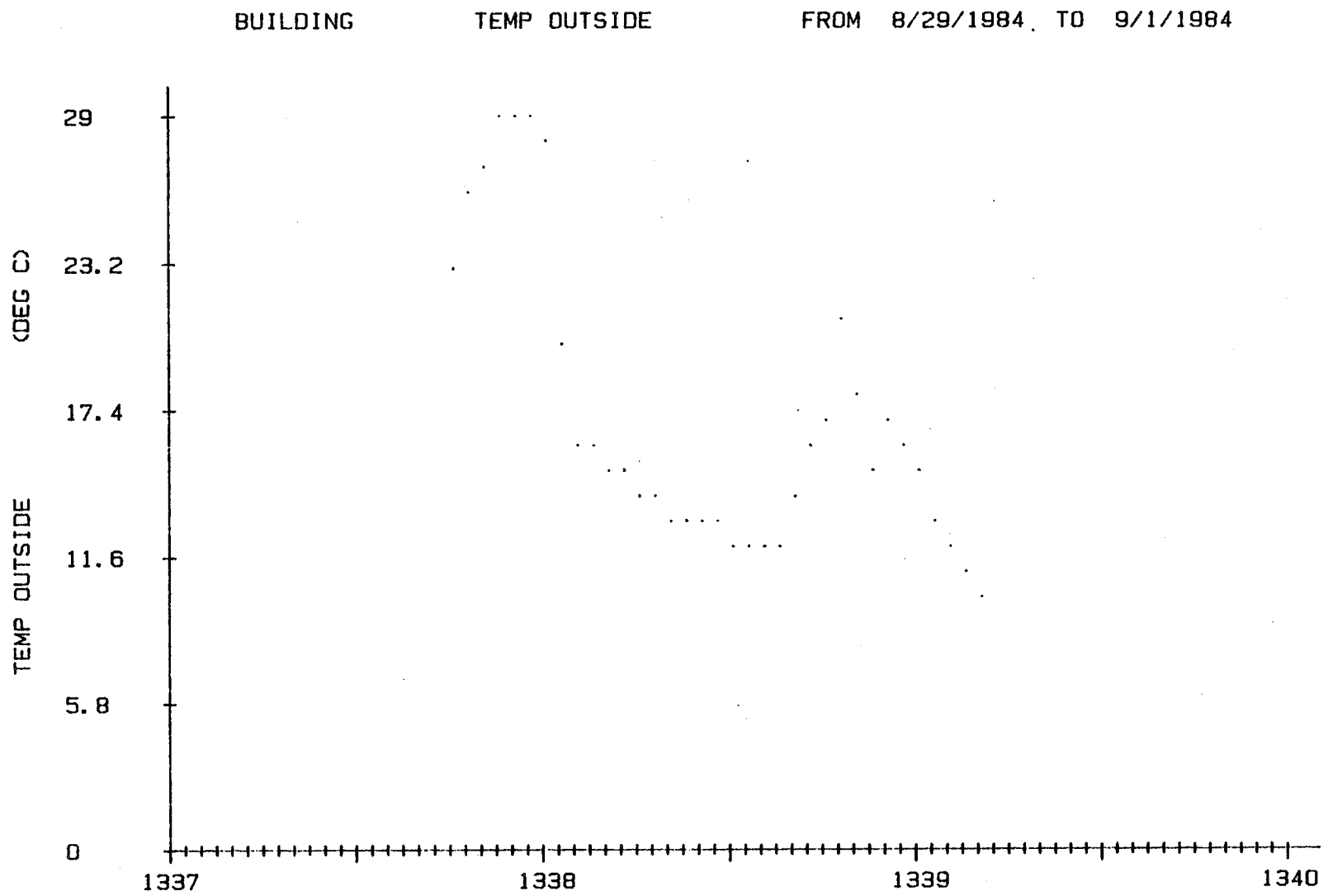

FIGURE A.1. Scatter Plot Generated by Option 'B' for Outside Temperature Data

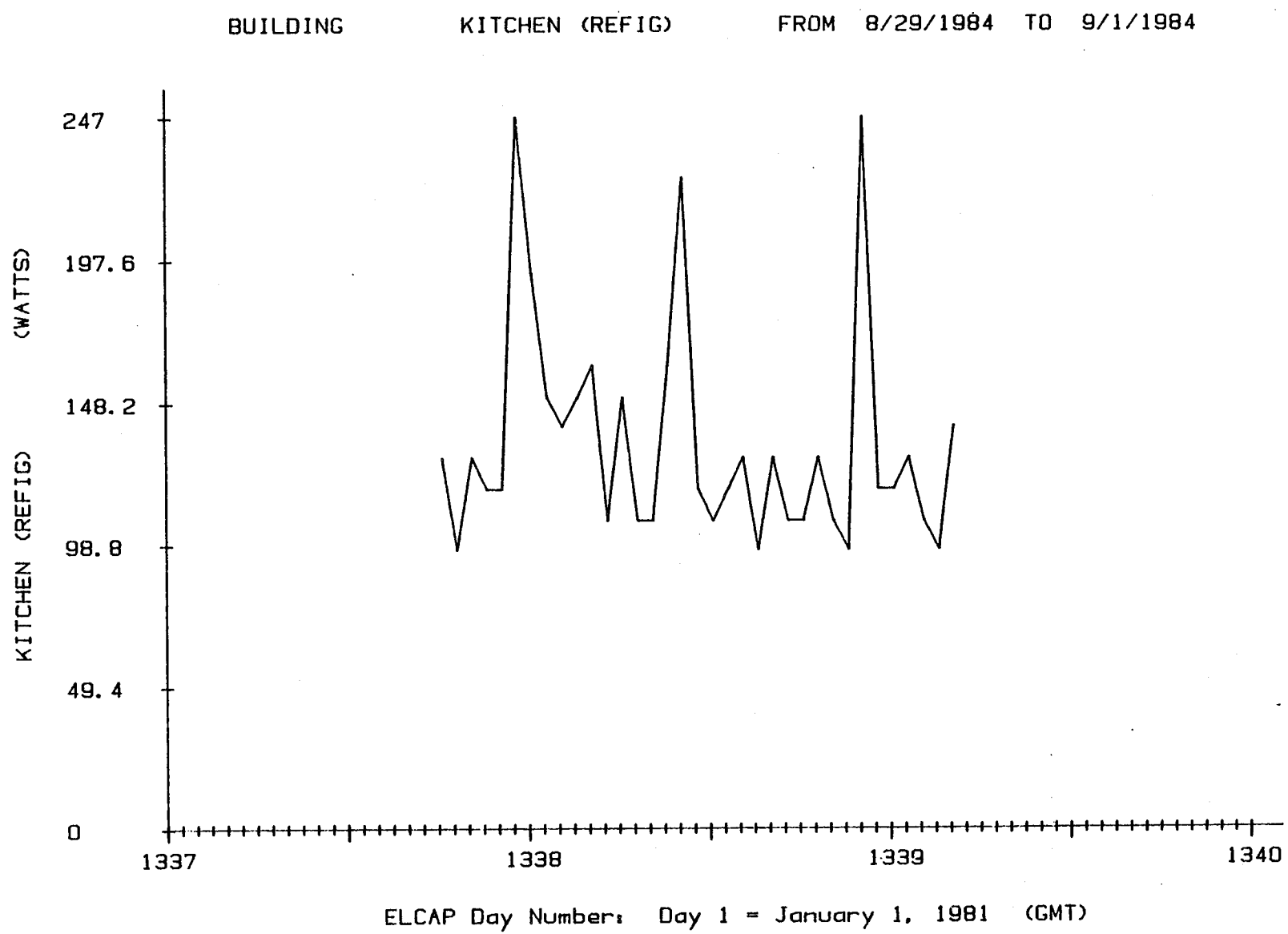

FIGURE A.2. Point-to-Point Plot Produced for the Refrigerator Using Option 'C'

TABLE A.10. Day Number Tables Produced by the Main Routine 'F'

1096--SUNDAY	JANUARY	1,	1984
1097--MONDAY	JANUARY	2,	1984
1098--TUESDAY	JANUARY	3,	1984
1099--WEDNESDAY	JANUARY	4,	1984
1100--THURSDAY	JANUARY	5,	1984
1101--FRIDAY	JANUARY	6,	1984
1102--SATURDAY	JANUARY	7,	1984
1103--SUNDAY	JANUARY	8,	1984
1104--MONDAY	JANUARY	9,	1984
1105--TUESDAY	JANUARY	10,	1984
1106--WEDNESDAY	JANUARY	11,	1984
1107--THURSDAY	JANUARY	12,	1984
1108--FRIDAY	JANUARY	13,	1984
1109--SATURDAY	JANUARY	14,	1984
1110--SUNDAY	JANUARY	15,	1984
1111--MONDAY	JANUARY	16,	1984
1112--TUESDAY	JANUARY	17,	1984
1113--WEDNESDAY	JANUARY	18,	1984
1114--THURSDAY	JANUARY	19,	1984
1115--FRIDAY	JANUARY	20,	1984
1116--SATURDAY	JANUARY	21,	1984
1117--SUNDAY	JANUARY	22,	1984
1118--MONDAY	JANUARY	23,	1984
1119--TUESDAY	JANUARY	24,	1984
1120--WEDNESDAY	JANUARY	25,	1984
1121--THURSDAY	JANUARY	26,	1984
1122--FRIDAY	JANUARY	27,	1984
1123--SATURDAY	JANUARY	28,	1984
1124--SUNDAY	JANUARY	29,	1984
1125--MONDAY	JANUARY	30,	1984
1126--TUESDAY	JANUARY	31,	1984
1127--WEDNESDAY	FEBRUARY	1,	1984
1128--THURSDAY	FEBRUARY	2,	1984
1129--FRIDAY	FEBRUARY	3,	1984
1130--SATURDAY	FEBRUARY	4,	1984
1131--SUNDAY	FEBRUARY	5,	1984
1132--MONDAY	FEBRUARY	6,	1984
1133--TUESDAY	FEBRUARY	7,	1984
1134--WEDNESDAY	FEBRUARY	8,	1984
1135--THURSDAY	FEBRUARY	9,	1984
1136--FRIDAY	FEBRUARY	10,	1984
1137--SATURDAY	FEBRUARY	11,	1984
1138--SUNDAY	FEBRUARY	12,	1984
1139--MONDAY	FEBRUARY	13,	1984
1140--TUESDAY	FEBRUARY	14,	1984
1141--WEDNESDAY	FEBRUARY	15,	1984
1142--THURSDAY	FEBRUARY	16,	1984
1143--FRIDAY	FEBRUARY	17,	1984
1144--SATURDAY	FEBRUARY	18,	1984
1145--SUNDAY	FEBRUARY	19,	1984
1146--MONDAY	FEBRUARY	20,	1984
1147--TUESDAY	FEBRUARY	21,	1984
1148--WEDNESDAY	FEBRUARY	22,	1984
1149--THURSDAY	FEBRUARY	23,	1984
1150--FRIDAY	FEBRUARY	24,	1984
1151--SATURDAY	FEBRUARY	25,	1984
1152--SUNDAY	FEBRUARY	26,	1984
1153--MONDAY	FEBRUARY	27,	1984
1154--TUESDAY	FEBRUARY	28,	1984
1155--WEDNESDAY	FEBRUARY	29,	1984
1156--THURSDAY	MARCH	1,	1984
1157--FRIDAY	MARCH	2,	1984
1158--SATURDAY	MARCH	3,	1984
1159--SUNDAY	MARCH	4,	1984
1160--MONDAY	MARCH	5,	1984
1161--TUESDAY	MARCH	6,	1984
1162--WEDNESDAY	MARCH	7,	1984
1163--THURSDAY	MARCH	8,	1984
1164--FRIDAY	MARCH	9,	1984
1165--SATURDAY	MARCH	10,	1984
1166--SUNDAY	MARCH	11,	1984
1167--MONDAY	MARCH	12,	1984
1168--TUESDAY	MARCH	13,	1984
1169--WEDNESDAY	MARCH	14,	1984
1170--THURSDAY	MARCH	15,	1984
1171--FRIDAY	MARCH	16,	1984
1172--SATURDAY	MARCH	17,	1984
1173--SUNDAY	MARCH	18,	1984
1174--MONDAY	MARCH	19,	1984
1175--TUESDAY	MARCH	20,	1984
1176--WEDNESDAY	MARCH	21,	1984
1177--THURSDAY	MARCH	22,	1984
1178--FRIDAY	MARCH	23,	1984
1179--SATURDAY	MARCH	24,	1984
1180--SUNDAY	MARCH	25,	1984
1181--MONDAY	MARCH	26,	1984
1182--TUESDAY	MARCH	27,	1984
1183--WEDNESDAY	MARCH	28,	1984
1184--THURSDAY	MARCH	29,	1984
1185--FRIDAY	MARCH	30,	1984

APPENDIX B

PROGRAM LISTING FOR TRANPLUS.BAS (SOURCE CODE)

```
A>TYPE TRANPLUS.BAS
10000 REM ************************** TRANZ **************************
10010 REM
10020 REM ********** FILE TRNSFORMATION AND EVALUATION PROGRAM **********
10030 CLEAR
10050 CLS : LOCATE 8,16 : PRINT "ENTER DISK DRIVE FOR BUILDING DATA FILES (A:, B:, etc.)"
10060 LOCATE 10,20 : INPUT "DISK MUST CONTAIN *.DAT' AND '.PAR' FILES", W$
10070 DIM B$(112)
10080 DEFINT D-E,G-L,N-O,T-Y,B
10090 DEFSNG A,M,F,S,C,Z
10100 DIM UNT$(112)
10110 DIM STREC(10)
10120 DIM ENDREC(10)
10130 DIM FDAY(2)
10140 DIM FSEC(2)
10150 DIM ADAY(2)
10160 DIM ASEC(2)
10170 DIM ATIME(2)
10180 DIM FTIME(2)
10190 DIM BADRECORD(255,9)
10200 DIM PARAMETER(36)
10210 DIM ZATA(3,255),ZCAL1(112),ZCAL2(112),ZAVE(112),ZMAX(112)
10220 DIM Y(4)
10230 DIM M(12)
10240 DIM IATA(255)
10250 DIM TI$(112)
10260 DIM D$(112)
10270 DIM CHAN$(112)
10280 DIM BIT$(112)
10290 DIM STATUS(112)
10300 DIM ZSCALE(112)
10310 DIM TEMPOR(5)
10320 DIM OVFL(999)
10330 DIM ARRAY(999,5)
10340 DIM RECORDA(1,255)
10350 CLS : LOCATE 6,1,0 : PRINT "          ***** MAIN MENU *****"
10360 PRINT
10370 PRINT "      KEY          CHOICE"
10380 PRINT "      ---          ------------------------------------------------"
10390 PRINT "       A           Convert Raw Data to an Engineering (ASCII) File"
10400 PRINT
10410 PRINT "       B           Check Engineering Data for Reasonableness"
10420 PRINT
10430 PRINT "       C           Sort an Engineering File Chronologically"
10440 PRINT
10450 PRINT "       D           Append Two Sorted Engineering Files"
10460 PRINT
10470 PRINT "       E           Summarize an Engineering File and Present Results"
10480 PRINT
10490 PRINT "       F           Generate Table of Day Numbers with Dates
10500 PRINT
10510 PRINT "       Q           QUIT ****"
10520 PRINT
10530 PRINT
10540 PRINT "      PRESS DESIRED KEY"
10550 A$ = INKEY$
10560 IF A$ = "" GOTO 10550
```

```
10570 IF A$ = "A" OR A$ = "a" GOTO 12500
10580 IF A$ = "B" OR A$ = "b" GOTO 16000
10590 IF A$ = "C" OR A$ = "c" GOTO 11000
10600 IF A$ = "D" OR A$ = "d" GOTO 14500
10610 IF A$ = "E" OR A$ = "e" GOTO 17500
10620 IF A$ = "F" OR A$ = "f" GOTO 22500
10630 IF A$ = "Q" OR A$ = "q" THEN END
10640 GOTO 10550
11000 REM
11010 REM **** SUBROUTINE TO SORT AN ENGINEERING FILE CHRONOLOGICALLY ****
11020 GOSUB 13630
11030 LOCATE 8,14 : INPUT "NAME OF ENGINEERING FILE TO BE SORTED"; EF$
11040 EFILE$ = W$ + EF$
11050 OPEN EFILE$ FOR INPUT AS #1
11060 LOCATE 12,25 : PRINT "ENTER NAME FOR NEW SORTED FILE:"
11070 LOCATE 13,25 : INPUT "(SAME AS ABOVE FOR WRITE-OVER)"; SRTD$
11080 CRATE! = 256*PARAMETER(5) + PARAMETER(6)
11090 SORTED$ = W$ + SRTD$
11100 CLS : LOCATE 6,1 : PRINT "PLEASE WAIT";
11110 REM
11120 REM **** SECTION TO READ FILE PARTS INTO AN ARRAY ****
11130 CC = 0
11140 FOR I = 1 TO 1000
11150    U = EOF(1) : IF U < 0 GOTO 11240
11160    LINE INPUT #1, X$
11170    ARRAY(I,1) = VAL(MID$(X$,1,3))
11180    FOR J = 2 TO 5
11190       ARRAY(I,J) = VAL(MID$(X$,4+(J-2)*6,6))
11200    NEXT J
11210    CC = CC + 1
11220    IF CC > 20 THEN PRINT "."; : CC = 0
11230 NEXT I
11240 CLOSE #1   RECNUM = I-1
11250 REM
11260 REM **** SECTION TO FIND THE FIRST ENTRY IN THE FILE ****
11270 PRINT "NOW SORTING FILE" : PRINT : PRINT : PRINT
11280 MINDAY = 10000 : MINSEC = 0 : MINREC = 0 : POW = 0 : NONO = 0
11290 FOR J = 1 TO I-1
11300    IF ARRAY(J,4) > MINDAY THEN GOTO 11340
11310    IF NONO = 0 AND ARRAY(J,4) < 100 THEN GOSUB 11900
11320    IF ARRAY(J,4) < MINDAY THEN MINDAY = ARRAY(J,4) : MINSEC = ARRAY(J,5) : MINREC = J
11330    IF ARRAY(J,5) < MINSEC THEN MINSEC = ARRAY(J,5) : MINREC = J
11340 NEXT J
11350 IF NONO = 0 THEN PRINT : PRINT POW;" POWER OUTAGES WERE DETECTED DURING SORT" : PRINT
11360 IF MINREC < 2 THEN PRINT : PRINT "FILE ALREADY SORTED" ELSE GOTO 11420
11370 PRINT : PRINT : PRINT "RESAVE WITH POWER OUTAGE DAYS/TIMES RESET (Y or N)?"
11380 K$ = INKEY$
11390 IF K$ = "" GOTO 11380
11400 IF K$ = "Y" OR K$ = "y" GOTO 11420
11410 IF K$ = "N" OR K$ = "n" THEN GOTO 10350 ELSE GOTO 11380
11420 REM
11430 REM **** SECTION TO ARRANGE AND SAVE THE FILE IN CHRONOLOGICAL ORDER ****
11440 IF POW > 0 THEN GOSUB 11960
11450 IF EFILE$ = SORTED$ THEN SORTED$ = W$ + "TEMPFIL"
11460 OPEN EFILE$ FOR INPUT AS #1
11470 OPEN SORTED$ FOR APPEND AS #2
11480 PRINT : PRINT : PRINT "PLEASE WAIT...";
11490 CC = 0
11500 FOR N = 1 TO MINREC-1
11510    CC = CC + 1
```

```
11520    IF CC ) 15 THEN PRINT " ";   CC = 0
11530    LINE INPUT #1, X$
11540 NEXT N
11550 INX = 0
11560 PRINT "SAVING SORTED FILE"
11570 FOR Q = MINREC TO RECNUM
11580     U = EOF(1) : IF U ( 0 GOTO 11700
11590     FOR VV = 1 TO POW
11600         IF Q ) STREC(VV)-1 AND Q ( ENDREC(VV)+1 THEN LINE INPUT #1, X$   GOTO 12220
11610     NEXT VV
11620     INX = INX + 1
11630     LINE INPUT #1, X$
11640     PRE$ = STR$(INX)
11650     IF INX ( 10 THEN INX$ = "  " + MID$(PRE$,2)
11660     IF INX ) 9 AND INX ( 100 THEN INX$ = " " + MID$(PRE$,2)
11670     IF INX ) 99 THEN INX$ = MID$(PRE$,2)
11680     MID$(X$,1,3) = INX$
11690     PRINT #2, USING "&"; X$
11700 NEXT Q
11710 CLOSE #1
11720 OPEN EFILE$ FOR INPUT AS #1
11730 FOR P = 1 TO MINREC-1
11740     FOR SS = 1 TO POW
11750         IF P ) STREC(SS)-1 AND P ( ENDREC(SS)+1 THEN LINE INPUT #1, X$ : GOTO 12320
11760     NEXT SS
11770     INX = INX + 1
11780     LINE INPUT #1, X$
11790     PRE$ = STR$(INX)
11800     IF INX ( 10 THEN INX$ = "  " + MID$(PRE$,2)
11810     IF INX ) 9 AND INX ( 100 THEN INX$ = " " + MID$(PRE$,2)
11820     IF INX ) 99 THEN INX$ = MID$(PRE$,2)
11830     MID$(X$,1,3) = INX$
11840     PRINT #2, USING "&"; X$
11850 NEXT P
11860 CLOSE #1
11870 CLOSE #2
11880 IF SORTED$ = V$ : "TEMPFIL" THEN KILL EFILE$ : NAME SORTED$ AS EFILE$
11890 GOTO 10350
11900 REM
11910 REM **** SUBROUTINE TO SET DELETE BOUNDARIES FOR POWER OUTAGES ****
11920 IF POW = 0 THEN SFD = 1 : STREC(SFD) = J : POW = 1 : ENDREC(SFD) = J   RETURN 11340
11930 IF J 1 ) ENDREC(SFD)  THEN SFD = SFD + 1   POW = POW + 1 : STREC(SFD) = J
11940 ENDREC(SFD) = J
11950 RETURN 11340
11960 REM
11970 REM **** SUBROUTINE TO RESET THE DAY AND SECONDS FOR A POWER OUTAGE ****
11980 IF STREC(1)=1 AND ENDREC(1)=RECNUM THEN PRINT : PRINT "DATA FOR POWER OUTAGE ONLY" : PRINT : MONO = 1 : POW = 0 : GOTO 11290
11990 SOUP = 1 : NUTS = POW   STP = 1
12000 IF ARRAY(STREC(POW),4)*86400!+ARRAY(STREC(POW),5) (= GRATE! AND ENDREC(POW) = RECNUM THEN SOUP = POW : NUTS = 1 : STP = -1
12010 FOR XX = SOUP TO NUTS STEP STP
12020     IF STREC(XX)-1 ( 1 THEN AAA = RECNUM ELSE AAA = STREC(XX)-1
12030     IF ENDREC(XX)+1 ) RECNUM THEN BBB = 1 ELSE BBB = ENDREC(XX)+1
12040     IF ARRAY(AAA,4)*86400!+ARRAY(AAA,5) ) ARRAY(BBB,4)*86400!+ARRAY(BBB,5) AND ARRAY(STREC(XX),4)*86400!+ARRAY(STREC(XX),5) (= GRATE! THEN GOSUB 12160 ELSE GOSUB 12140
12050 NEXT XX
12060 FOR RR = 1 TO POW
12070     IF ARRAY(STREC(RR),4)*86400!+ARRAY(STREC(RR),5) ( ARRAY(MINREC,4)*86400!+ARRAY(MINREC,5) THEN MINREC = STREC(RR)
12080 NEXT RR
12090 RETURN
12100 FOR WW = STREC(XX) TO ENDREC(XX)
12110     IF WW-1 ( 1 THEN CCC = RECNUM ELSE CCC = WW-1
```

```
12120     ARRAY(VV,4) = ARRAY(CCC,4) : ARRAY(VV,5) = ARRAY(CCC,5) + CRATE!
12130     IF ARRAY(VV,5) ) 86400! THEN ARRAY(VV,4) = ARRAY(VV,4) + 1 : ARRAY(VV,5) = ARRAY(VV,5) - 86400!
12140 NEXT VV
12150 RETURN
12160 FOR VV = ENDREC(XX) TO STREC(XX) STEP -1
12170     IF VV )= RECNUM THEN DDD = 1 ELSE DDD = VV+1
12180     ARRAY(VV,4) = ARRAY(DDD,4) : ARRAY(VV,5) = ARRAY(DDD,5) - CRATE!
12190     IF ARRAY(VV,5) ( 0 THEN ARRAY(VV,4) = ARRAY(VV,4) - 1 : ARRAY(VV,5) = 86400! + ARRAY(VV,5)
12200 NEXT VV
12210 RETURN
12220 REM
12230 REM **** SECTION TO PREP POWER OUTAGE FOR RESAVE ****
12240 AAAA$ = STR$(ARRAY(Q,4))
12250 APASS = ARRAY(Q,4) : GOSUB 12400
12260 MID$(I$,16,4) = AAAA$
12270 AAAA$ = STR$(ARRAY(Q,5))
12280 APASS = ARRAY(Q,5) : GOSUB 12400
12290 MID$(I$,22,4) = AAAA$
12300 INX = INX + 1
12310 GOTO 11640
12320 AAAA$ = STR$(ARRAY(P,4))
12330 APASS = ARRAY(P,4) : GOSUB 12400
12340 MID$(I$,16,4) = AAAA$
12350 AAAA$ = STR$(ARRAY(P,5))
12360 APASS = ARRAY(P,5) : GOSUB 12400
12370 MID$(I$,22,4) = AAAA$
12380 INX = INX + 1
12390 GOTO 11790
12400 IF APASS ( 10 THEN AAAA$ = "     " + MID$(AAAA$,2,1)
12410 IF APASS ( 100 AND APASS ) 9 THEN AAAA$ = "    " + MID$(AAAA$,2,2)
12420 IF APASS ( 1000 AND APASS ) 99 THEN AAAA$ = "   " + MID$(AAAA$,2,3)
12430 IF APASS ( 10000 AND APASS ) 999 THEN AAAA$ = "  " + MID$(AAAA$,2,4)
12440 IF APASS ( 100000! AND APASS ) 9999 THEN AAAA$ = " " + MID$(AAAA$,2,5)
12450 IF APASS )= 100000! THEN AAAA$ = " 99999"
12460 RETURN
12500 REM
12510 REM *** SUBROUTINE TO CONVERT RAW DATA TO AN ENGINEERING (ASCII) FILE ***
12520 GOSUB 11630
12530 LOCATE 21,1
12540 INPUT "COMPRESSED FORMAT FILE NAME"; F8$
12550 F$ = W$ + F8$
12560 INTSTR=1
12570 PRINT "NUMBER OF CHANNELS IS ";INTEND
12580 INTREC=1
12590 INRECED=RECNUM
12600 INPUT "ASCII OUTPUT DATA FILE NAME";FFIL$
12610 FFILE$ = W$ + FFIL$
12620 CRATE! = 256*PARAMETER(5) + PARAMETER(6)
12630 PRINT
12640 PRINT
12650 PRINT "        PLEASE WAIT"
12660 PRINT
12670 PRINT "        TAKE TIME OUT TO GO GET A CUP OF COFFEE OR WHATEVER YOU LIKE"
12680 PRINT
12690 OPEN F$ AS #3 LEN=1
12700 FIELD 3,1 AS PAR$
12710 GET #3,5
12720 LENGTH=ASC(PAR$)
12730 GET #3,1
12740 L1=ASC(PAR$)
```

```
12750 GET #3,2
12760 L2=ASC(PAR$)
12770 RNUM=(L1*256)+L2
12780 GOSUB 13140
12790 OPEN FILE$ FOR APPEND AS #1
12800 FOR I=INTREC TO IRECID
12810    IF I>RNUM THEN GOTO 12840
12820    GOSUB 12910
12830    GOSUB 13090
12840    GOSUB 13460
12850    GOSUB 13490
12860 NEXT I
12870 CLOSE #3 : CLOSE #1
12880 GOTO 10350
12890 REM
12900 REM **** READS ONE LINE OF BYTES INTO AN ARRAY ****
12910 K=LENGTH*(I-1)+2
12920 FOR J=1 TO LENGTH
12930    K=K+1
12940    GET #3,K
12950    RECORDA(1,J)=ASC(PAR$)
12960 NEXT J
12970 RETURN
12980 REM
12990 REM **** PREPARES UNCALIBRATED ENGINEERING (ASCII) VALUES FOR ONE LINE ****
13000 ZATA(1,1)=RECORDA(1,1)*256+RECORDA(1,2) : IATA(1)=0
13010 ZATA(1,2)=RECORDA(1,3) : IATA(2)=1
13020 ZATA(1,3)=RECORDA(1,4)*256+RECORDA(1,5) : IATA(3)=0
13030 Z1=RECORDA(1,6) : Z2=RECORDA(1,7) : Z3=RECORDA(1,8)
13040 ZATA(1,4)=Z1*65536!+Z2*256+Z3
13050 ZATA(1,5)=RECORDA(1,9) : IATA(5)=1
13060 K=5 : J=10
13070 FOR II=1 TO 112
13080    IF STATUS(II)=0 THEN GOTO 13110
13090    K=K+1
13100    IF VAL(BIT$(II))=0 THEN ZATA(1,K)=RECORDA(1,J)*256+RECORDA(1,J+1) : IATA(K)=0 : J=J+2 : ELSE ZATA(1,K)=RECORDA(1,J) : IATA(K)=1 : J=J+1
13110  NEXT II
13120 FLENGTH=K
13130 RETURN
13140 REM
13150 REM **** SUBROUTINE TO GET VALUES FROM 'DAT' FILE ****
13160 GOSUB 13570
13170 K = 6
13180 FOR J=1 TO 112
13190    IF STATUS(J)=0 THEN GOTO 13260
13200    GET #2,J
13210    ZCAL1(K)=VAL(CL1$)
13220    ZCAL2(K)=VAL(CL2$)
13230    ZMAX(K)=VAL(MV$)
13240    IF AVE$="F" THEN ZAVE(K) = CRATE! ELSE ZAVE(K)=1
13250    K=K+1
13260 NEXT J
13270 CLOSE #1
13280 OPEN PARFILE$ AS #2 LEN=4   FIELD #2,4 AS PAR1$
13290 K=6
13300 FOR J=1 TO 112
13310    IF STATUS(J)=0 THEN GOTO 13350
13320    GET #2,J+35
13330    ZSCALE(K)=2^VAL(PAR1$)
13340    K=K+1
```

```
13350 NEXT J
13360 CLOSE #2
13370 RETURN
13380 REM
13390 REM **** SETS THE ACTUAL ENGINEERING (ASCII) VALUES FOR ONE LINE ****
13400 K=4
13410 FOR J=1 TO 112
13420    IF STATUS(J)=0 THEN GOTO 13450
13430    ZATA(1,K)=(ZATA(1,K)*ZSCALE(K)*ZCAL1(K)/ZAVE(K))+ZCAL2(K)
13440    K=K+1
13450 NEXT J
13460 RETURN
13470 REM
13480 REM **** WRITES ONE LINE INTO THE ENGINEERING (ASCII) FILE ****
13490 PRINT #1, USING "###";I.
13500 FOR J=INTSTR TO INTEND
13510   PRINT #1, USING " #####";ZATA(1,J);
13520 NEXT J
13530 PRINT #1," "
13540 RETURN
13550 REM
13560 REM **** SUBROUTINE TO OPEN 'DAT' FILE ****
13570 F1$= W$ + BUILDING$ + ".DAT"
13580 OPEN F1$ AS #2 LEN=56
13590 FIELD #2, 20 AS TL$, 12 AS CL1$, 12 AS CL2$, 5 AS UT$, 1 AS AVE$, 4 AS MV$
13600 RETURN
13610 REM
13620 REM **** SECTION TO INPUT BUILDING AND SET CHANNEL STATUS ****
13630 CLS   LOCATE 12,30 : INPUT "BUILDING"; BUILDING$
13640 PARFILE$ = W$ + BUILDING$ + ".PAR"
13650 OPEN PARFILE$ AS #2 LEN=4
13660 FIELD #2,4 AS PAR$
13670 FOR I = 1 TO 35
13680    GET #2,I
13690    PARAMETER(I-1)=VAL(PAR$)
13700 NEXT I
13710 CLOSE #2
13720 LENGTH=9
13730 FIRST = 7    LAST = 20 : GOSUB 13900
13740 FOR I = 1 TO 112
13750    IF BIT$(I)= "1" THEN STATUS(I)=1 ELSE STATUS(I)=0
13760 NEXT I
13770 Z=0
13780 FOR I = 1 TO 112
13790   IF STATUS(I)=1 THEN Z=Z+1
13800 NEXT I
13810 INTEND = Z+5
13820 FIRST = 21 : LAST = 34 : GOSUB 13900
13830 FOR I=1 TO 112
13840    IF STATUS(I)=1 THEN LENGTH =LENGTH +1 + VAL(BIT$(I))
13850 NEXT I
13860 LOCATE 13,30   PRINT "RECORD LENGTH "; LENGTH
13870 RECNUM=FIX(15360/LENGTH)+1
13880 FLEN = INTEND*6 + 3
13890 RETURN
13900 FOR K=FIRST TO LAST
13910    D-PARAMETER(K)
13920    GOSUB 13980
13930    FOR L = 1 TO 8
13940       BIT$((K-FIRST)*8+L)=B$(7-L)
```

```
13950    NEXT L
13960 NEXT K
13970 RETURN
13980 FOR H=8 TO 1 STEP -1
13990    H1=H-1
14000    IF D)=2^H1 THEN D=D-(2^H1) : B$(H)="1" ELSE B$(H)="0"
14010 NEXT H
14020 LET B$(1)= RIGHT$(STR$(D),1)
14030 RETURN
14500 REM
14510 REM **** SUBROUTINE TO APPEND TWO SORTED FILES ****
14520 CLS : LOCATE 6,16   INPUT "NAME OF FIRST FILE"; FIRFIL$
14530 FIRFIL$ = W$ + FIRFIL$
14540 LOCATE 9,16 : INPUT "NAME OF FILE TO APPEND", APPFIL$
14550 APPFIL$ = W$ + APPFIL$
14560 LOCATE 12,16 : INPUT "NAME FOR NEWLY APPENDED OUTPUT FILE"; PUTFIL$
14570 PUTFIL$ = W$ + PUTFIL$
14580 REM
14590 REM **** SECTION TO DETERMINE BEGINNING AND ENDING TIMES IN FILES ****
14600 CLS : LOCATE 4,1 : PRINT "READING DATA FROM FILES...";
14610 OPEN FIRFIL$ FOR INPUT AS #1 : OPEN APPFIL$ FOR INPUT AS #2
14620 LINE INPUT #1, FIRLIN$ : LINE INPUT #2, APPLIN$
14630 FDAY(1) = VAL(MID$(FIRLIN$,16,6)) : FSEC(1) = VAL(MID$(FIRLIN$,23,6))
14640 ADAY(1) = VAL(MID$(APPLIN$,16,6))  ASEC(1) = VAL(MID$(APPLIN$,23,6))
14650 CC = 0
14660 FOR VV = 2 TO 1000
14670    CC = CC + 1 : IF CC ) 25 THEN PRINT " "; : CC = 0
14680    U = EOF(1) : IF U ( 0 GOTO 14710
14690    LINE INPUT #1, FIRLIN$
14700 NEXT VV
14710 FOR WW = 2 TO 1000
14720    CC = CC + 1   IF CC ) 25 THEN PRINT " "; : CC = 0
14730    U = EOF(2)  IF U ( 0  GOTO 14760
14740    LINE INPUT #2, APPLIN$
14750 NEXT WW
14760 PRINT "COMPLETE"
14770 RNUMF = VV : RNUMA = WW
14780 FDAY(2) = VAL(MID$(FIRLIN$,16,6)) : FSEC(2) = VAL(MID$(FIRLIN$,23,6))
14790 ADAY(2) = VAL(MID$(APPLIN$.16,6)) : ASEC(2) = VAL(MID$(APPLIN$,23,6))
14800 CLOSE #1 : CLOSE #2
14810 ATIME(1) = ADAY(1)*86400! + ASEC(1) : ATIME(2) = ADAY(2)*86400! + ASEC(2)
14820 FTIME(1) = FDAY(1)*86400! + FSEC(1) : FTIME(2) = FDAY(2)*86400! + FSEC(2)
14830 REM
14840 REM **** SECTION TO DETERMINE FILE RELATIONSHIPS AND TO BRANCH ****
14850 IF FTIME(1) = ATIME(1) AND FTIME(2) = ATIME(2) GOTO 15540
14860 IF FTIME(1) = ATIME(1) AND FTIME(2) ) ATIME(2) GOTO 15480
14870 IF FTIME(2) = ATIME(2) AND FTIME(1) ( ATIME(1) GOTO 15480
14880 IF FTIME(1) ( ATIME(1) AND FTIME(2) ) ATIME(2) GOTO 15480
14890 IF ATIME(1) = FTIME(1) AND ATIME(2) ) FTIME(2) GOTO 15540
14900 IF ATIME(2) = FTIME(2) AND ATIME(1) ( FTIME(1) GOTO 15540
14910 IF ATIME(1) ( FTIME(1) AND ATIME(2) ) FTIME(2) GOTO 15540
14920 IF FTIME(2) ( ATIME(1) THEN STT$ = "I" : GOTO 15560
14930 IF ATIME(2) ( FTIME(1) THEN STT$ = "M" : GOTO 15580
14940 IF ATIME(1) ) FTIME(1) AND ATIME(1) (= FTIME(2) AND ATIME(2) ) FTIME(2) GOTO 15740
14950 REM
14960 REM **** SECTION TO DETERMINE WHERE FILES OVERLAP ****
14970 CUTFIL$ = FIRFIL$ : RNUM = RNUMF : STAFIL$ = APPFIL$ : OLDREC = RNUMA : STATIME = ATIME(2)
14980 OPEN CUTFIL$ FOR INPUT AS #1
14990 FOR JJ = 1 TO 1000
15040    U = EOF(1)  IF U ( 0 GOTO 15540
```

```
15010     LINE INPUT #1, FLN$
15020     CHKDAY = VAL(MID$(FLN$,16,6))
15030     CHKSEC = VAL(MID$(FLN$,22,6))
15040     CHKTIME = CHKDAY*86400! + CHKSEC
15050     IF CHKTIME > STATIME GOTO 15090
15060 NEXT JJ
15070 REM
15080 REM **** SECTION TO CREATE THE OUTPUT FILE ****
15090 NEWREC = RNUM - JJ + 1 + OLDREC
15100 CLS : LOCATE 10,1 : PRINT "CREATING NEW APPENDED FILE...";
15110 OPEN STAFIL$ FOR INPUT AS #2
15120 OPEN PUTFIL$ FOR APPEND AS #3
15130 CC = 0
15140 FOR II = 1 TO OLDREC
15150 PRE$ = STR$(II)
15160     IF II < 10 THEN IN$ = "  " + MID$(PRE$,2)
15170     IF II > 9 AND II < 100 THEN IN$ = " " + MID$(PRE$,2)
15180     IF II > 99 THEN IN$ = MID$(PRE$,2)
15190     CC = CC + 1 : IF CC > 15 THEN PRINT ".";  : CC = 0
15200     U = EOF(2) : IF U < 0 GOTO 15240
15210     LINE INPUT #2, SLIN$
15220     MID$(SLIN$,1,3) = IN$
15230     PRINT #3, USING "&"; SLIN$
15240 NEXT II
15250 IF U = 0 THEN II = II + 1
15260 PRE$ = STR$(II)
15270 IF II < 10 THEN IN$ = "  " + MID$(PRE$,2)
15280 IF II > 9 AND II < 100 THEN IN$ = " " + MID$(PRE$,2)
15290 IF II > 99 THEN IN$ = MID$(PRE$,2)
15300 MID$(FLN$,1,3) = IN$
15310 PRINT #3, USING "&", FLN$
15320 FOR HH = II + 1 TO NEWREC
15330     CC = CC + 1 : IF CC > 15 THEN PRINT ".";  : CC = 0
15340     U = EOF(1) : IF U < 0 GOTO 15420
15350     LINE INPUT #1, FLN$
15360     PRE$ = STR$(HH)
15370     IF HH < 10 THEN IN$ = "  " + MID$(PRE$,2)
15380     IF HH < 100 AND HH > 9 THEN IN$ = " " + MID$(PRE$,2)
15390     IF HH > 99 THEN IN$ = MID$(PRE$,2)
15400     MID$(FLN$,1,3) = IN$
15410     PRINT #3, USING "&"; FLN$
15420 NEXT HH
15430 PRINT "DONE" : SOUND 32767,27 : SOUND 32767,.1
15440 CLOSE #1 : CLOSE #2 : CLOSE #3
15450 GOTO 10350
15460 REM
15470 REM **** MESSAGES AND INITIALIZATIONS FOR VARIOUS FILE CONFIGURATIONS ****
15480 CLS : LOCATE 6,1 : PRINT "TIME PERIOD FOR ";FIRFIL$;" INCLUDES ALL OF ";APPFIL$
15490 LOCATE 10,1 : PRINT "FILES NOT APPENDED"
15500 PRINT : PRINT : PRINT "PRESS ANY KEY TO CONTINUE"
15510 K$ = INKEY$
15520 IF K$ = "" GOTO 15510
15530 GOTO 10350
15540 CLS   LOCATE 6,1   PRINT "TIME PERIOD FOR ";APPFIL$;" INCLUDES ALL OF ";FIRFIL$
15550 GOTO 15490
15560 CLS : LOCATE 6,1   PRINT "FILES ";FIRFIL$;" AND ";APPFIL$;" INCLUDE THE SAME TIME PERIOD"
15570 GOTO 15490
15580 CLS : LOCATE 6,1 : PRINT "FILES ";FIRFIL$;" AND ";APPFIL$;" INCLUDE TOTALLY DIFFERENT TIME PERIODS"
15590 LOCATE 9,1   PRINT "FILES MAY BE DISJOINT"
15600 LOCATE 12,1 : PRINT "CONTINUE TO APPEND (Y or N)?"
```

```
15610 K$ = INKEY$
15620 IF K$ = "" GOTO 15610
15630 IF K$ = "Y" AND STT$ = "Y" OR K$ = "y" AND STT$ = "Y" GOTO 15730
15640 IF K$ = "Y" OR K$ = "y" GOTO 15670
15650 IF K$ = "N" OR K$ = "n" GOTO 10350
15660 GOTO 15610
15670 CUTFIL$ = FIRFIL$ : STAFIL$ = APPFIL$ : OLDREC = RNUMA
15680 OPEN CUTFIL$ FOR INPUT AS #1
15690 LINE INPUT #1, FLN$
15700 NEWREC = RNUMA + RNUMF
15710 CLS : LOCATE 10,1 : PRINT "CREATING NEW APPENDED FILE...",
15720 GOTO 15110
15730 CUTFIL$ = APPFIL$ : STAFIL$ = FIRFIL$ : OLDREC = RNUMF : GOTO 15680
15740 CUTFIL$ = APPFIL$ : RNUM = RNUMA : STAFIL$ = FIRFIL$ : OLDREC = RNUMF : STATIME = FTIME(2)
15750 GOTO 14980
16000 REM
16010 REM **** SUBROUTINE TO CHECK FOR DATA REASONABLENESS ****
16020 CLS : LOCATE 10,30  PRINT "RESETTING ARRAYS"
16030 GOSUB 22010
16040 GOSUB 13630
16050 FOR ORG = 1 TO 112
16060     ZAVE(ORG) = 0 : ZCAL1(ORG) = 0
16070 NEXT ORG
16080 GOSUB 13570
16090 CLS: LOCATE 8,14 : INPUT "NAME OF ENGINEERING FILE TO BE CHECKED";EF$
16100 EFILE$ = W$ + EF$
16110 LOCATE 11,14 : INPUT "MINIMUM VALUE FOR TEMPERATURE DATA"; LLL
16120 LOCATE 13,14 : INPUT "MAXIMUM VALUE FOR TEMPERATURE DATA"; HHH
16130 IF LLL = 0 AND HHH = 0 THEN LLL = -50 : HHH = 60
16140 IF LLL >= HHH THEN LLL = -50 : HHH = 60
16150 STT$ = ""
16160 CLS : LOCATE 8,14 : PRINT "DO YOU WANT A HARD COPY OF THE RESULTS (Y or N)?"
16170 K$ = INKEY$
16180 IF K$ = "" GOTO 16170
16190 IF K$ = "Y" OR K$ = "y" THEN STT$ = "Y" : GOTO 16220
16200 IF K$ = "N" OR K$ = "n" GOTO 16220
16210 GOTO 16170
16220 OPEN EFILE$ FOR INPUT AS #1
16230 LINE INPUT #1, CHKL$
16240 CLOSE #1
16250 N = 1 : BEGIN = 34 : OATT = 4 : EC = 0 : HH = 0 : RCHK = 0 : WARN = 0 : OVF = 0 : IF = 0 : OCHK = 0 : BDAY = 10000 : EDAY = 0
16260 CLS : LOCATE 10,25 : PRINT "READING FROM '.DAT' FILE"
16270 GOSUB 17070
16280 CLS : LOCATE 10,30 : PRINT "CHECKING FILE"
16290 PRINT : PRINT : PRINT "PLEASE WAIT...";
16300 FOR TIT = 1 TO FIX((LEN(CHKL$)-34)/61+1
16310     PRINT " ";
16320     OPEN EFILE$ FOR INPUT AS #1
16330     FOR II = 1 TO 1000
16340         LINE INPUT #1, EFLIN$
16350         IF VAL(MID$(EFLIN$,16,6)) < BDAY THEN BDAY = VAL(MID$(EFLIN$,16,6))
16360         IF VAL(MID$(EFLIN$,16,6)) > EDAY THEN EDAY = VAL(MID$(EFLIN$,16,6))
16370         FOR JJ = 0 TO OATT
16380             STUD$ = MID$(EFLIN$,BEGIN+OVFL(II)+6*JJ,6)
16390             IF MID$(STUD$,2,1) = "%" THEN GOSUB 17300
16400             ARRAY(II,JJ+1) = VAL(STUD$)
16410         NEXT JJ
16420         U = EOF(1) : IF U < 0 THEN RECNUM = II : CLOSE #1 : GOTO 16440
16430     NEXT II
16440     VV = 0
```

```
16450     PRINT ".",
16460     FOR J = N TO N + OATT
16470         VV = VV + 1
16480         IF ZMAX(J) (= 0 THEN GOSUB 17360
16490         IF MID$(TI$(J),1,4) = "TEMP" THEN GOSUB 17420
16500         FOR R = 1 TO RECNUM
16510             IF ARRAY(R,VV) ( 0 OR ARRAY(R,VV) ) ZMAX(J) THEN GOSUB 17180
16520         NEXT R
16530     NEXT J
16540     N = N + 5   BEGIN = BEGIN + 30
16550     IF LEN(CHKL$) - BEGIN ( 30 THEN OATT = FIX((LEN(CHKL$)-BEGIN)/4) -1
16560     IF LEN(CHKL$) - BEGIN ( 30 AND FIX((LEN(CHKL$)-BEGIN)/6) ( ((LEN(CHKL$)-BEGIN)/6 THEN OATT = FIX((LEN(CHKL$)-BEGIN)/4)
16570     IF OATT ( 0 GOTO 16590
16580 NEXT TTT
16590 PRINT "DONE" : SOUND 32767,9 : SOUND 32767,.1
16600 FLEAT = BDAY : GOSUB 22800 : CLS : LOCATE 4,20 : PRINT "DATA SPANS THE TIME PERIOD FROM:" : LOCATE 4,28 : PRINT USING "00";MO; : PRINT "/"; : PRINT USING "00";DAY; : PRINT "/"; : PRINT USING "0000";YR;
16610 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "DATA SPANS THE TIME PERIOD FROM:   "; : LPRINT USING "00";MO; : LPRINT "/"; : LPRINT USING "00";DAY; : LPRINT "/"; : LPRINT USING "0000";YR;
16620 FLEAT = EDAY : GOSUB 22800 : PRINT "    TO    "; : PRINT USING "00";MO; : PRINT "/"; : PRINT USING "00";DAY; : PRINT "/"; : PRINT USING "0000";YR
16630 IF STT$ = "Y" THEN LPRINT "    TO    "; : LPRINT USING "00";MO; : LPRINT "/"; : LPRINT USING "00";DAY; : LPRINT "/"; : LPRINT USING "0000";YR
16640 PRINT : PRINT : PRINT "PRESS ANY KEY TO CONTINUE"
16650 K$ = INKEY$
16660 IF K$ = "" GOTO 16650
16670 IF EC = 0 THEN PRINT : PRINT "        NO OUT OF RANGE VALUES FOUND"
16680 IF EC = 0 AND STT$ = "Y" THEN LPRINT : LPRINT "         NO OUT OF RANGE VALUES FOUND" : GOTO 16770
16690 CLS : LOCATE 4,25 : PRINT "******* SUMMARY OF ERRORS *******" : PRINT : PRINT
16700 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "          ******* SUMMARY OF ERRORS *******" : LPRINT : LPRINT
16710 FOR VV = 1 TO HH
16720     PRINT BIT$(VV); TAB(45); ZAVE(VV); TAB(52); "ERRORS" : PRINT
16730     IF STT$ = "Y" THEN LPRINT BIT$(VV);TAB(45);ZAVE(VV);TAB(52);"ERRORS" : LPRINT
16740 NEXT VV
16750 PRINT : PRINT : PRINT "TOTAL NUMBER OF OUT OF RANGE VALUES IS ";EC
16760 IF STT$ = "Y" THEN LPRINT : LPRINT "TOTAL NUMBER OF OUT OF RANGE VALUES IS ",EC
16770 PRINT : PRINT : PRINT "PRESS ANY KEY TO CONTINUE"
16780 K$ = INKEY$
16790 IF K$ = "" GOTO 16780
16800 IF OVF = 0 THEN PRINT : PRINT "         NO OVERFLOWS DETECTED" : SOUND 32767,45   SOUND 32767,.1
16810 IF OVF = 0 AND STT$ = "Y" THEN LPRINT : LPRINT "         NO OVERFLOWS DETECTED" : GOTO 16900
16820 CLS : LOCATE 4,25 : PRINT "******* SUMMARY OF OVERFLOWS *******" : PRINT : PRINT
16830 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "          ******* SUMMARY OF OVERFLOWS *******" : LPRINT : LPRINT
16840 FOR YY = 1 TO FF
16850     PRINT B$(YY),TAB(45) ZCALL(YY);TAB(53), "OVERFLOWS" : PRINT
16860     IF STT$ = "Y" THEN LPRINT B$(YY);TAB(45);ZCALL(YY),TAB(53);"OVERFLOWS" : LPRINT
16870 NEXT YY
16880 PRINT : PRINT : PRINT "TOTAL NUMBER OF OVERFLOWS =";OVF
16890 IF STT$ = "Y" THEN LPRINT : LPRINT "TOTAL NUMBER OF OVERFLOWS =";OVF
16900 PRINT : PRINT "PRESS ANY KEY TO CONTINUE"
16910 K$ = INKEY$
16920 IF K$ = "" GOTO 16910
16930 IF WARN = 0 THEN PRINT : PRINT : PRINT "        NO WARNINGS FOUND"
16940 CLS : LOCATE 4,25 : PRINT "******* SUMMARY OF WARNINGS *******" : PRINT : PRINT
16950 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "          ******* SUMMARY OF WARNINGS *******" : LPRINT : LPRINT
16960 FOR UU = 1 TO WARN
16970     PRINT "MAXIMUM VALUE FOR ";D$(UU), TAB(40); "LESS THAN OR EQUAL TO ZERO" : PRINT
16980     IF STT$ = "Y" THEN LPRINT "MAXIMUM VALUE FOR ";D$(UU);TAB(40);"LESS THAN OR EQUAL TO ZERO" : LPRINT
16990 NEXT UU
17000 PRINT : PRINT : PRINT "TOTAL NUMBER OF WARNINGS IS ";WARN
17010 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "TOTAL NUMBER OF WARNINGS IS ";WARN
17020 PRINT : PRINT : PRINT "PRESS ANY KEY TO CONTINUE"
17030 K$ = INKEY$
17040 IF K$ = "" GOTO 17030
```

```
17050 CLOSE #2
17060 GOTO 10350
17070 REM
17080 REM **** SUBROUTINE TO READ VALUES FOR "DAT" FILE ****
17090 Q = 0
17100 FOR I = 1 TO 112
17110     IF STATUS (I) = 0 THEN GOTO 17140
17120     Q = Q + 1
17130     GET #2, I
17140     TI$(Q) = TL$   UNT$(Q) = UT$
17150     ZMAX(Q) = VAL(CL1$) * VAL(MV$) + VAL(CL2$)
17160 NEXT I
17170 RETURN
17180 REM
17190 REM **** SUBROUTINE TO NOTE THE ERRORS FOUND ****
17200 EC = EC + 1
17210 IF RCHK () J THEN HH = HH + 1 : BIT$(HH) = TI$(J)
17220 ZAVE(HH) = ZAVE(HH) + 1
17230 RCHK = J
17240 IF ARRAY(R,VV) ( 99999! GOTO 17290
17250 OVF = OVF + 1
17260 IF OCHK () J THEN FF = FF + 1 : BS(FF) = TI$(J)
17270 ZCAL1(FF) = ZCAL1(FF) + 1
17280 OCHK = J
17290 RETURN
17300 REM
17310 REM **** SUBROUTINE TO TRAP PRINTING OVERFLOW ERRORS ****
17320 CHK$ = MID$(EFLIN$,BEGIN+OVFL(II)+6*JJ+6,1)
17330 IF CHK$ () " " THEN OVFL(II) = OVFL(II) + 1 : GOTO 17320
17340 STUD$ = " 99999"
17350 RETURN
17360 REM
17370 REM **** SUBROUTINE FOR SETTING WARNING PARAMETERS ****
17380 WARN = WARN + 1
17390 DS(WARN) = TI$(J)
17400 IF TI$(J) = "" THEN DS(WARN) = "** NO LABEL **"
17410 RETURN
17420 REM
17430 REM **** SUBROUTINE TO CHECK THE RANGE OF TEMPERATURES ****
17440 FOR R = 1 TO RECNUM
17450     IF ARRAY(R,VV) ( LLL AND ARRAY(R,VV) ) HHH THEN GOSUB 17360
17460 NEXT R
17470 RETURN 16530
17500 REM
17510 REM **** SUBROUTINE TO SUMMARIZE DATA AND DISPLAY RESULTS ****
17520 GOSUB 13430
17530 GOSUB 13570
17540 GOSUB 17080
17550 CRATE! = 256*PARAMETER(5)+PARAMETER(6)
17560 CLOSE #2
17570 GOSUB 19190
17580 CLS : LOCATE 6,10 : PRINT "***** SUMMARY MENU *****"
17590 PRINT
17600 PRINT "      KEY         CHOICE"
17610 PRINT "      ---         ------------------------------------------------"
17620 PRINT "       A          Tabular Summaries (Min, Max, Avg, Std Dev, Total)"
17630 PRINT
17640 PRINT "       B          Scatter Plot of Usage Data "
17650 PRINT
17660 PRINT "       C          Point-to-Point Plot of Usage Data"
```

```
17670 PRINT
17680 PRINT "        P          Display 'ON CHANNEL' Descriptions"
17690 PRINT
17700 PRINT "        Q          QUIT **** (Return to Main Menu)"
17710 PRINT : PRINT
17720 PRINT "        PRESS DESIRED KEY"
17730 SPP$ = ""
17740 A$ = INKEY$
17750 IF A$ = "" GOTO 17740
17760 IF A$ = "A" OR A$ = "a" GOTO 18006
17770 IF A$ = "B" OR A$ = "b" GOTO 20000
17780 IF A$ = "C" OR A$ = "c" THEN SPP$ = "Y" : GOTO 20000
17790 IF A$ = "P" OR A$ = "p" THEN GOSUB 19190
17800 IF A$ = "Q" OR A$ = "q" GOTO 18350
17810 GOTO 17580
18000 REM
18010 REM **** SUBROUTINE TO CREATE A TABULAR SUMMARY ****
18020 CLS : LOCATE 4,1 : PRINT "NAME OF ENGINEERING FILE TO BE SUMMARIZED (DO NOT ENTER 'P' FOR DESCRIPTIONS)" : INPUT EFILE$
18030 EFILE$ = W$ + EFILE$
18040 STT$ = ""
18050 CLS : LOCATE 4,16 : PRINT "DO YOU WANT A HARD COPY OF THE TABLE (Y or N)?"
18060 K$ = INKEY$
18070 IF K$ = "" GOTO 18060
18080 IF K$ = "Y" OR K$ = "y" THEN STT$ = "Y"   GOTO 18110
18090 IF K$ = "N" OR K$ = "n" GOTO 18110
18100 GOTO 18060
18110 CLS : PRINT : INPUT "NUMBER OF CHANNELS TO BE SUMMARIZED AND TABULATED (1-ALL)";CHANUM$
18120 IF CHANUM$ = "P" OR CHANUM$ = "p" THEN GOSUB 19190 : GOTO 18110
18130 IF CHANUM$ = "ALL" OR CHANUM$ = "all" GOTO 19480
18140 CHANUM = VAL(CHANUM$)
18150 CLS : PRINT : PRINT "ENTER 'ON CHANNEL' NUMBERS AS GIVEN IN THE TABLE (OR 'P' FOR DESCRIPTION DISPLAY)"
18160 FOR YY = 1 TO CHANUM
18170     INPUT CHAN$(YY)
18180     IF CHAN$(YY) = "P" OR CHAN$(YY) = "p" THEN GOSUB 19190 : GOTO 18150
18190     ZATA(1,YY) = VAL(CHAN$(YY))
18200     CHAN$(YY) = TI$(ZATA(1,YY))
18210     IF CHAN$(YY) = "" THEN CHAN$(YY) = "** NO LABEL **"
18220 NEXT YY
18230 FOR JJ = 1 TO CHANUM
18240     IF ZATA(1,JJ) > 0 AND ZATA(1,JJ) <= 0 GOTO 18260
18250     ZATA(1,JJ) = 0   ZMAX(JJ) = 0 : ZSCALE(JJ) = 0 : ZAVE(JJ) = 0
18260 NEXT JJ
18270 BEGIN = 34   OATT = 4
18280 GOSUB 22010
18299 FOR DDD = 1 TO 112
18300     ZCAL1(DDD) = 0   ZCAL2(DDD) = 0
18310 NEXT DDD
18320 OPEN EFILE$ FOR INPUT AS #1
18330 LINE INPUT #1, CHKL$
18340 SDAY = VAL(MID$(CHKL$,16,6))
18350 CLOSE #1
18360 REM
18370 REM **** MAIN LOOP FOR SUMMARIZING DATA ****
18380 CLS : LOCATE 6,6 : PRINT "PLEASE WAIT"
18390 LOCATE 9,6 : PRINT "NOW SUMMARIZING CHANNELS  .",
18400 FOR TTT = 1 TO FIX((LEN(CHKL$)-34)/6)+1
18410     OPEN EFILE$ FOR INPUT AS #1
18420     PRINT " ",
18430     FOR II = 1 TO 1000
18440         LINE INPUT #1, EFLIN$
```

```
18450       FOR JJ = 0 TO OATT
18460          STUD$ = MID$(EFLIN$,BEGIN+OVFL(II)+5*JJ,6)
18470          IF MID$(STUD$,1,1) = "%" THEN GOSUB 17360
18480          ARRAY(II,JJ+1) = VAL(STUD$)
18490       NEXT JJ
18500       U = EOF(1) : IF U < 0 THEN RECNUM = II : FDAY = VAL(MID$(EFLIN$,14,6)) : CLOSE #1 : GOTO 18520
18510    NEXT II
18520    PRINT ".";
18530    FOR GG = 1 TO 5
18540       FOR EE = 1 TO CHANUM
18550          IF (BEGIN-34)/6 + GG = ZATA(1,EE) THEN GOSUB 19050
18560       NEXT EE
18570    NEXT GG
18580    BEGIN = BEGIN + 30
18590    IF LEN(CHKL$)-BEGIN < 30 THEN OATT = FIX((LEN(CHKL$)-BEGIN)/6)-1
18600    IF LEN(CHKL$)-BEGIN < 30 AND FIX((LEN(CHKL$)-BEGIN)/6) < (LEN(CHKL$)-BEGIN)/6 THEN OATT = FIX((LEN(CHKL$)-BEGIN)/6)
18610    IF OATT < 0 GOTO 18640
18620    PRINT ".";
18630 NEXT TTT
18640 PRINT "DONE"
18650 REM
18660 REM **** SUBROUTINE TO PRINT OUT THE SUMMARIES ****
18670 FLEAT = SDAY : GOSUB 22810
18680 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "ELCAP DAY NUMBER   (DAY 1 = JANUARY 1, 1981)-- ";SDAY;" TO ";FDAY : LPRINT
18690 IF STT$ = "Y" THEN LPRINT : LPRINT "DATES INCLUDE:  "; : LPRINT USING "00";MO; : LPRINT "/"; : LPRINT USING "00";DAY; : LPRINT "/"; : LPRINT USING "0000";YR; : LPRINT "    TO    ";
18700 CLS : LOCATE 6,32 : PRINT BUILDING$
18710 LOCATE 9,15 : PRINT "ELCAP DAY NUMBER   (DAY 1 = JANUARY 1, 1981)-- ";SDAY;" TO ";FDAY
18720 LOCATE 11,15 : PRINT "DATES INCLUDE:  "; : PRINT USING "00";MO; : PRINT "/"; : PRINT USING "00";DAY; : PRINT "/"; : PRINT USING "0000";YR; : PRINT "    TO    ";
18730 FLEAT = FDAY : GOSUB 22810
18740 IF STT$ = "Y" THEN LPRINT USING "00";MO;  LPRINT "/"; : LPRINT USING "00";DAY; : LPRINT "/"; : LPRINT USING "0000";YR
18750 PRINT USING "00";MO; : PRINT "/"; : PRINT USING "00";DAY; : PRINT "/"; : PRINT USING "0000";YR
18760 PRINT : PRINT : PRINT "PRESS ANY KEY TO CONTINUE"
18770 K$ = INKEY$
18780 IF K$ = "" THEN GOTO 18770
18790 DC = 1 : DD = 10
18800 IF CHANUM < DD THEN DD = CHANUM
18810 CLS : LOCATE 1,15 : PRINT "SUMMARIES FOR CHANNELS";DC;"TO";DD;"      ";BUILDING$ : PRINT
18820 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "SUMMARIES FOR CHANNELS";DC,"TO";DD;"      ";BUILDING$ : LPRINT
18830 PRINT "ON CHANNEL",TAB(26);"UNITS",TAB(33);"MINIMUM";TAB(42);"MAXIMUM";TAB(51);"AVERAGE";TAB(60);"STAND DEV";TAB(71);"TOT (W-H)" : PRINT
18840 IF STT$ = "Y" THEN LPRINT "ON CHANNEL";TAB(26);"UNITS";TAB(33);"MINIMUM";TAB(42);"MAXIMUM";TAB(51);"AVERAGE";TAB(60);"STAND DEV";TAB(71);"TOT (W-H)" : LPRINT
18850 FOR TNT = DC TO DD
18860    PRINT ZATA(1,TNT);TAB(5);CHAN$(TNT);TAB(26);UNT$(ZATA(1,TNT));TAB(33);
18870    PRINT USING "0000000"; ZATA(2,TNT); : PRINT TAB(42); : PRINT USING "0000000"; ZMAX(TNT); : PRINT TAB(51); : PRINT USING "0000000"; ZAVE(TNT); : PRINT TAB(60); : PRINT USING "00000.000"; ZSCALE(TNT); : PRINT TAB(
70),
18880    PRINT USING "000.0*****"; ZCAL1(TNT) : PRINT
18890    IF STT$ = "Y" THEN LPRINT ZATA(1,TNT);TAB(5);CHAN$(TNT);TAB(26);UNT$(ZATA(1,TNT));TAB(33);
18900    IF STT$ = "Y" THEN LPRINT USING "0000000"; ZATA(2,TNT); : LPRINT TAB(42); : LPRINT USING "0000000"; ZMAX(TNT); : LPRINT TAB(51); : LPRINT USING "0000000"; ZAVE(TNT); : LPRINT TAB(60); : LPRINT USING "0000.000";
ZSCALE(TNT); : LPRINT TAB(70),
18910    IF STT$ = "Y" THEN LPRINT USING "000.0*****"; ZCAL1(TNT) : LPRINT
18920 NEXT TNT
18930 PRINT "PRESS ANY KEY TO CONTINUE"
18940 K$ = INKEY$
18950 IF K$ = "" GOTO 18940
18960 IF CHANUM = DD THEN GOSUB 18970 : GOTO 17580
18970 DD = DD + 10 : DC = DC + 10
18980 GOTO 18800
18990 REM
19000 REM **** SUBROUTINE TO FREE STRING SPACE ****
19010 FOR BB = 1 TO CHANUM
19020    CHAN$(BB) = ""
```

```
19030 NEXT BB
19040 RETURN
19050 REM
19060 REM **** SUBROUTINE TO CALCULATE STATS FOR A GIVEN CHANNEL ****
19070 ZATA(2,EE) = 99999!   ZMAX(EE) = 0
19080 FOR H = 1 TO RECNUM
19090     IF ARRAY(H,GG) < ZATA(2,EE) THEN ZATA(2,EE) = ARRAY(H,GG)
19100     IF ARRAY(H,GG) > ZMAX(EE) THEN ZMAX(EE) = ARRAY(H,GG)
19110     ZCAL1(EE) = ZCAL1(EE) + ARRAY(H,GG)
19120     ZCAL2(EE) = ZCAL2(EE) + ARRAY(H,GG) * ARRAY(H,GG)
19130 NEXT H
19140 ZSCALE(EE) = (ZCAL2(EE)-ZCAL1(EE)*ZCAL1(EE)/RECNUM)/(RECNUM-1)
19150 IF ZSCALE(EE) < 0 THEN ZSCALE(EE) = 0 : GOTO 19170
19160 ZSCALE!(EE) = SQR(ZSCALE(EE))
19170 ZAVE(EE) = ZCAL1(EE)/RECNUM
19180 RETURN
19190 REM
19200 REM **** SUBROUTINE TO DISPLAY 'ON CHANNEL' DESCRIPTIONS ****
19210 STT$ = ""
19220 CLS : LOCATE 4,4 : PRINT "DO YOU WANT A HARD COPY OF 'ON CHANNEL' DESCRIPTIONS AND NUMBERS (Y/N)?"
19230 K$ = INKEY$
19240 IF K$ = "" GOTO 19230
19250 IF K$ = "Y" OR K$ = "y" THEN STT$ = "Y" : GOTO 19280
19260 IF K$ = "N" OR K$ = "n" GOTO 19280
19270 GOTO 19230
19280 DC = 1   DD = 10
19290 IF Q < DD THEN DD = Q
19300 CLS : LOCATE 1,25 : PRINT "DESCRIPTIONS FOR 'ON CHANNELS'";DC;"TO";DD : PRINT
19310 IF STT$ = "Y" THEN LPRINT : LPRINT : LPRINT "DESCRIPTIONS FOR 'ON CHANNELS'";DC;"TO";DD : LPRINT
19320 FOR TNT = DC TO DD
19330     PRINT "ON CHANNEL #";TNT;TAB(30);TI$(TNT) : PRINT
19340     IF STT$ = "Y" THEN LPRINT "ON CHANNEL #";TNT;TAB(30);TI$(TNT) : LPRINT
19350 NEXT TNT
19360 PRINT "PRESS ANY KEY TO CONTINUE"
19370 K$ = INKEY$
19380 IF K$ = "" GOTO 19370
19390 IF Q = DD GOTO 19420
19400 DD = DD + 10 : DC = DC + 10
19410 GOTO 19290
19420 CLS : LOCATE 8,12 : PRINT "TO RETURN TO CHANNEL DESCRIPTIONS,"
19430 LOCATE 10,7 : PRINT "PRESS P AT NEARLY ANY PAUSE IN THE SUMMARY ROUTINE"
19440 PRINT   PRINT : PRINT "PRESS ANY KEY TO CONTINUE "
19450 K$ = INKEY$
19460 IF K$ = "" GOTO 19450
19470 CLS : RETURN
19480 FOR NNN = 1 TO Q
19490     ZATA(1,NNN) = NNN
19500     CHAN$(NNN) = TI$(NNN)
19510     IF CHAN$(NNN) = "" THEN CHAN$(NNN) = "** NO LABEL **"
19520 NEXT NNN
19530 CHANUM = Q
19540 GOTO 16230
20000 REM
20010 REM **** SUBROUTINE TO GENRATE A SCATTER OR POINT-TO-POINT PLOT ****
20020 CLS   LOCATE 4,16 : PRINT "DO YOU HAVE A GRAPHICS CAPABLE SCREEN (Y/N)?"
20030 K$ = INKEY$
20040 IF K$ = "" GOTO 20030
20050 IF K$ = "Y" OR K$ = "y" THEN STS = 1   GOTO 20080
20060 IF K$ = "N" OR K$ = "n" THEN STS = 0 : GOTO 20080
20070 GOTO 20030
```

```
20080 CLS : LOCATE 4,14 : INPUT "NAME OF SORTED ENGINEERING FILE TO BE PLOTTED";EFILE$
20090 EFILE$ = W$ + EFILE$
20100 ZZZ = FRE("")
20110 CLS : LOCATE 6,16  INPUT "ON CHANNEL NUMBER FOR PLOT TO BE CREATED";CHAN$
20120 IF CHAN$ = "P" OR CHAN$ = "p" THEN GOSUB 19190 : GOTO 20110
20130 CHAN = VAL(CHAN$)
20140 CHKK = 0
20150 CHKK2 = 0
20160 CLS : LOCATE 2,20 : PRINT "ELCAP DAY 1 = JANUARY 1, 1981"
20170 LOCATE 4,1 : INPUT "ELCAP DAY NUMBER AT WHICH PLOT WILL BEGIN ('ALL' FOR FILE)"; STPLOT$
20180 IF STPLOT$ = "ALL" OR STPLOT$ = "all" THEN CHKK2 = 1 : GOTO 20220 ELSE STPLOT = VAL(STPLOT$)
20190 LOCATE 7,6 : INPUT "TIME (IN SECONDS) FOR THE ABOVE DAY";ZZZ : STPLOT = STPLOT*86400! + ZZZ
20200 LOCATE 11,1: INPUT "ELCAP DAY NUMBER AT WHICH PLOT WILL END"; ENPLOT
20210 LOCATE 14,6 : INPUT "TIME (IN SECONDS) FOR THE ABOVE DAY";ZZZ : ENPLOT = ENPLOT*86400! + ZZZ
20220 CLS : LOCATE 6,6 : PRINT "PLEASE WAIT"
20230 LOCATE 9,6 : PRINT "GATHERING DATA FOR PLOT...";
20240 BEGIN = 34 : DDAY = 0
20250 GOSUB 22010
20260 OPEN EFILE$ FOR INPUT AS #1
20270 PRINT ".."; : DF = 0
20280 FOR II = 1 TO 1000
20290     DF = DF + 1 : IF DF > 20 THEN PRINT "."; : DF = 0
20300     LINE INPUT #1, EFLIN$
20310     IF CHKK2 = 1 GOTO 20340
20320     IF VAL(MID$(EFLIN$,14,6))*86400! + VAL(MID$(EFLIN$,22,6)) < STPLOT GOTO 20480
20330     IF VAL(MID$(EFLIN$,14,6))*86400! + VAL(MID$(EFLIN$,22,6)) > ENPLOT THEN CLOSE #1 : RECNUM = II - 1 : GOTO 20500
20340     IF CHKK = 0 THEN GOGO = II
20350     IF CHKK = 0 AND VAL(MID$(EFLIN$,22,6)) = 0 THEN DDAY = DDAY + 1 : DATA(1) = II
20360     IF CHKK = 0 THEN OLDDAY = VAL(MID$(EFLIN$,14,6)) : STIME = OLDDAY*86400! + VAL(MID$(EFLIN$,22,6))
20370     CHKK = 1
20380     ARRAY(II,3) = VAL(MID$(EFLIN$,14,6))
20390     ARRAY(II,4) = VAL(MID$(EFLIN$,22,6))
20400     ARRAY(II,2)= ARRAY(II,3)*86400! + ARRAY(II,4)
20410         FOR JJ = 0 TO CHAN-1
20420             STUD$ = MID$(EFLIN$,BEGIN+OVFL(II)+6*JJ,6)
20430             IF MID$(STUD$,2,1) = "%" THEN GOSUB 17300
20440         NEXT JJ
20450         ARRAY(II,1) = VAL(STUD$)
20460     IF OLDDAY < ARRAY(II,3) THEN DDAY = DDAY + 1 : OLDDAY = ARRAY(II,3) : DATA(DDAY) = II
20470     U = EOF(1) : IF U < 0 THEN CLOSE #1 : GOTO 20490
20480 NEXT II
20490 RECNUM = II
20500 PRINT "COMPLETE"
20510 IF ARRAY(GOGO,2) = ARRAY(RECNUM,2) THEN PRINT : PRINT : PRINT "NO DATA IN CHOSEN TIME PERIOD.... . PLOT ABANDONED" . SOUND 440,9 : SOUND 32767,4.5 : SOUND 440,9 : SOUND 32767,45 : SOUND 32767,.1 : GOTO 17580
20520 IF ARRAY(GOGO,2) > ARRAY(RECNUM,2) THEN PRINT : PRINT : PRINT "FILE NOT SORTED.... ..PLOT ABANDONED" : SOUND 440,9 : SOUND 32767,4.5 : SOUND 440,9 : SOUND 32767,63 : SOUND 32767,.1 : GOTO 17580
20530 ZMIN = 0 : ZMAX = 0
20540 FOR TT = GOGO TO RECNUM
20550     IF ARRAY(TT,1) > ZMAX THEN ZMAX = ARRAY(TT,1)
20560     IF ARRAY(TT,1) < ZMIN THEN ZMIN = ARRAY(TT,1)
20570 NEXT TT
20580 IF ZMAX = 0 AND ZMIN = 0 THEN PRINT  PRINT : PRINT "ALL VALUES EQUAL TO ZERO"  SOUND 440,9 : SOUND 32767,4.5 : SOUND 440,9 : SOUND 32767,45 : SOUND 32767,.1 : GOTO 21960
20590 HTH = 250 * ABS(ZMIN)/(ABS(ZMIN)+ZMAX)
20600 SCALE = 250/(ABS(ZMIN)+ZMAX)
20610 IF FIX(ARRAY(GOGO,2)/86400!) <> ARRAY(GOGO,2)/86400! THEN STIME = FIX(ARRAY(GOGO,2)/86400!)*86400!
20620 IF FIX(ARRAY(RECNUM,2)/86400!) <> ARRAY(RECNUM,2)/86400! THEN ETIME =(FIX(ARRAY(RECNUM,2)/86400!)+1)*86400!
20630 IF STS = 0 GOTO 20730
20640 SCREEN 2
20650 CLS
20660 FOR C = 1 TO (ETIME-STIME)/86400!+1
20670     AT =(86400!*(C-1)*350)/(ETIME-STIME)
```

```
20680     IF AT ) 350 GOTO 20720
20690     AT = (AT+50)*(650/415)
20700     BB1 = 199 - (HTH+64)*(200/325) : BB2 = 199 - (HTH+57)*(200/325)
20710     LINE (AT,BB1)-(AT,BB2)
20720 NEXT G
20730 AXIS = 0
20740 IF HTH )= 50 THEN AXIS = 50
20750 IF HTH )= 100 THEN AXIS = 100
20760 IF HTH )= 150 THEN AXIS = 150
20770 IF HTH )= 200 THEN AXIS = 200
20780 IF HTH )= 250 THEN AXIS = 250
20790 IF STS = 0 GOTO 20970
20800 HTH2 = 199 - (HTH+60)*(200/325)
20810 FOR N = 0 TO 5
20820     PLACE = 50*N+HTH-AXIS : IF PLACE ) 250 GOTO 20850
20830     PLACE = 199 - (PLACE+60)*(200/325)
20840     LINE (72.4819,PLACE)-(81.7349,PLACE)
20850 NEXT N
20860 LINE (645.8675,HTH2)-(77.1084,HTH2)
20870 LINE (77.1084,0.2308)-(77.1084,142.0769)
20880 FOR H = GOGO TO RECNUM
20890     AX =350*(ARRAY(H,2)-STIME)/(MTIME-STIME)
20900     AX = (AX+50)*(640/415)
20910     AY = SCALE*ARRAY(H,1) + HTH
20920     AY = 199 - (AY+60)*(200/325)
20930     IF SPP$ = "Y" AND H ) GOGO THEN LINE -(AX,AY)
20940     PSET (AX,AY)
20950 NEXT H
20960 LOCATE 1,11  PRINT"KEY TO CONTINUE";TAB(34);TI$(CHAN);TAB(55);UNT$(CHAN);TAB(67);BUILDING$
20970 FOR K = 1 TO 6
20980     ZCAL2(K) =(50*(K-1)-AXIS)*(ABS(ZMIN)+ZMAX)/(250)
20990 NEXT K
21000 IF STS = 0 GOTO 21110
21010 PP = 0
21020 FOR LL = 1 TO 6
21030     IF PP = 2 OR ZCAL2(LL) ) ZMAX GOTO 21090
21040     PP = 21+6*(AXIS-HTH)/50-(LL-1)*6
21050     IF PP (= 1 THEN PP = 2
21060     NUM$ = MID$(STR$(ZCAL2(LL)),1,6)
21070     LOCATE PP,2
21080     PRINT USING "&"; NUM$
21090 NEXT LL
21100 LOCATE 22,9 : PRINT USING "####"; STIME/86400!
21110 STP = 1
21120 IF DDAY ) 5 THEN STP = 2
21130 IF DDAY ) 10 THEN STP = 3
21140 IF DDAY ) 15 THEN STP = 4
21150 IF DDAY ) 20 THEN STP = 5
21160 IF DDAY ) 25 THEN STP = 6
21170 IF DDAY ) 30 THEN STP = 7
21180 IF STS = 0 GOTO 21270
21190 FOR P = 0 TO DDAY STEP STP
21200     BB = 9 + 60*(ARRAY(IATA(P),2)-STIME)/(MTIME-STIME)
21210     IF BB ( 14 GOTO 21250
21220     IF BB ) 72 GOTO 21250
21230     LOCATE 12,BB
21240     PRINT USING "####", ARRAY(IATA(P),3)
21250 NEXT P
21260 LOCATE 22,77 : PRINT USING "####"; MTIME/86400!
21270 FLEAT = STIME/86400! : GOSUB 22010  MO1 = MO : DAY1 = DAY : YR1 = YR
```

```
21280 FLEAT = MTIME/86400! : GOSUB 21810
21290 IF STS = 0 GOTO 21360
21300 LOCATE 24,3 : PRINT "ELCAP DAY NUMBER:  DAY 1 = JANUARY 1,.1981 (GMT)     ";
21310 PRINT USING "##",MO1; : PRINT "/"; : PRINT USING "##";DAY1; : PRINT "/"; : PRINT USING "####";YR1; : PRINT " TO ";
21320 PRINT USING "##",MO; : PRINT "/"; : PRINT USING "##";DAY; : PRINT "/";  PRINT USING "####",YR;
21330 K$ = INKEY$
21340 IF K$ = "" GOTO 21330
21350 PRINT : SCREEN 0
21360 CLS : LOCATE 4,14 : PRINT "DO YOU WANT A HARD COPY OF THE PLOT (Y/N)?"
21370 K$ = INKEY$
21380 IF K$ = "" GOTO 21370
21390 IF K$ = "N" OR K$ = "n" THEN STP = 0 : GOTO 21960
21400 IF K$ = "Y" OR K$ = "y" GOTO 21420
21410 GOTO 21370
21420 CLS : LOCATE 4,10 : PRINT "DO YOU HAVE AN HP PLOTTER? (A = 7475A, B = 7470A, N = NONE) "
21430 K$ = INKEY$
21440 IF K$ = "" GOTO 21430
21450 IF K$ = "A" OR K$ = "a" THEN SPL = 1 : GOTO 21490
21460 IF K$ = "B" OR K$ = "b" THEN SPL = 2 : GOTO 21490
21470 IF K$ = "N" OR K$ = "n" GOTO 21960
21480 GOTO 21430
21490 CLS : LOCATE 4,14 : PRINT "LOAD PAPER FOR PLOT AND PREPARE PLOTTER" : LOCATE 9,14 : PRINT "PRESS ANY KEY TO CONTINUE"
21500 K$ = INKEY$
21510 IF K$ = "" GOTO 21500
21520 CLS : LOCATE 4,14 : INPUT "COMMUNICATIONS PORT NUMBER (COM1, COM2)"; CCM$ : PLOTTER$ = CCM$ + ":9600,S,7,1,RS,CS65535,DS,CD"
21530 OPEN PLOTTER$ AS #1
21540 IF SPL = 1 THEN CLS : INPUT "PAPER SIZE FOR PLOT? (1 = 11 X 17,  2 = 8.5 X 11)";SIZE : PRINT #1, "PS",SIZE+2,";"
21550 CLS : LOCATE 6,10 : PRINT "PLEASE WAIT..." : LOCATE 9,10 : PRINT "NOW GENERATING PLOT"
21560 PRINT #1, "IN;SC-40,375,-30,295;SP1;PU0,260;PD0,0;PU360,";HTH;";PD0,";HTH;",PU0,0;SP2;" : SOUND 32767,225 : SOUND 32767,.1
21570 FOR H = GOGO TO RECNUM
21580     AX = 350*(ARRAY(H,2)-STIME)/(MTIME-STIME)
21590     AY = SCALE*ARRAY(H,1) + HTH
21600     IF SPP$ = "Y" AND H ) GOGO THEN PRINT #1, "PD",AX,",",AY,";PU;"
21610     PRINT #1, "PU",AX,",",AY,";PD;PU;"
21620     SOUND 32767,12 : SOUND 32767,.1
21630 NEXT H
21640 PRINT #1, "SP1;" : SOUND 32767,75 : SOUND 32767,.1
21650 IF (GRATE ( 301 AND (MTIME-STIME)/86400! ( 5) OR (MTIME-STIME)/86400! ( 4 THEN MSTEP=350/(((MTIME-STIME)/86400!)*24) : TICKS=((MTIME-STIME)/86400!)*24+1 : TIC=1 : ELSE TICKS=(MTIME-STIME)/86400!+1 : MSTEP=350/((MTIM
E-STIME)/86400!) : TIC=0
21660 FOR G = 1 TO TICKS
21670     IF TIC = 1 AND (G MOD 12) = 1 THEN PRINT #1, "TL.5,1.25;PU",MSTEP*(G-1),",",HTH,";PD;XT;TL.5,.5;" ELSE PRINT #1, "PU",MSTEP*(G-1),",",HTH,";PD;XT;"
21680     SOUND 32767,4 : SOUND 32767,.1
21690 NEXT G
21700 SOUND 32767,35 : SOUND 32767,.1
21710 FOR N = 1 TO 6
21720     PLACE = 50*(N-1)+HTH-AXIS   IF PLACE ) 250 GOTO 21750
21730     PRINT #1, "PU0,",PLACE,";PD;XT;PU",DUMMY-60,",",PLACE-2.5,";LB",STR$(ZCAL2(N)),CHR$(3),";"
21740     SOUND 32767,45 : SOUND 32767,.1
21750 NEXT N
21760 PRINT #1, "PU",DUMMY-60,",",DUMMY-13,";LB",MID$(STR$(STIME/86400!),2,5),CHR$(3),";"
21770 FOR P = 1 TO DDAY STEP STP
21780     BB = 350*(ARRAY(IATA(P),2)-STIME)/(MTIME-STIME)-45
21790     IF BB ( -40 OR BB ) 265 GOTO 21830
21800     PRINT #1, "PU",BB,",",DUMMY-13,";LB",STR$(ARRAY(IATA(P),3)),CHR$(3),";"
21810     SOUND 32767,45 : SOUND 32767,.1
21820 NEXT P
21830 PRINT #1, "PU",DUMMY+286,",",DUMMY-13,";LB",STR$(MTIME/86400!),CHR$(3),";
21840 SOUND 32767,150 : SOUND 32767,.1
21850 PRINT #1, "PU3,280,LB",BUILDING$,"  ";TI$(CHAN)," FROM  ";MID$(STR$(MO1),2,2);"/";MID$(STR$(DAY1),2,2);"/";MID$(STR$(YR1),2,4);"  TO  ";MID$(STR$(MO),2,2);"/";MID$(STR$(DAY),2,2);"/";MID$(STR$(YR),2,4);CHR$(3),";"
21860 SOUND 32767,150 : SOUND 32767,.1
```

```
21870 PRINT #1, "PU70,-35;LBELCAP Day Number:  Day 1 = January 1, 1981  (GMT)"+CHR$(3) : SOUND 32767,180 : SOUND 32767,.1
21880 PRINT #1, "PU-44,0;DI0,1;LB",TI$(CHAN);"  (";UNT$(CHAN);")",CHR$(3),";"
21890 CLOSE #1
21900 CLS : LOCATE 4,14 : PRINT "WANT ANOTHER ORIGINAL OF THE SAME PLOT (Y/N)?"
21910 K$ = INKEY$
21920 IF K$ = "" GOTO 21910
21930 IF K$ = "Y" OR K$ = "y" GOTO 21490
21940 IF K$ = "N" OR K$ = "n" GOTO 21960
21950 GOTO 21910
21960 CLS : LOCATE 4,14 : PRINT "PLOT ANOTHER CHANNEL FROM THE SAME FILE (Y or N)?"
21970 K$ = INKEY$
21980 IF K$ = "Y" OR K$ = "y" GOTO 20110
21990 IF K$ = "N" OR K$ = "n" GOTO 17580
22000 GOTO 21970
22010 REM
22020 REM **** SUBROUTINE TO RESET THE OVERFLOW COUNTERS ****
22030 FOR ORG = 1 TO 999
22040     OVFL(ORG)=0
22050 NEXT ORG
22060 RETURN
22500 REM
22510 REM **** SUBROUTINE TO CREATE TABLE OF DAY NUMBERS AND CORRESPONDING DATES ****
22520 CLS : LOCATE 6,3 : PRINT "(JANUARY 1, 1981 = 1)    (JANUARY 1,1982 = 366)    (JANUARY 1, 1983 = 731)"
22530 LOCATE 8,1 : PRINT "(JANURAY 1, 1984 = 1096)    (JANUARY 1, 1985 = 1462)   (JANUARY 1, 1986 = 1827)"
22540 LOCATE 4,14 : INPUT "ELCAP DAY NUMBER TO START THE TABLE";PDAY1
22550 LOCATE 10,14 : INPUT "ELCAP DAY NUMBER TO END THE TABLE";PDAY2
22560 SKIP = 61
22570 SECT = FIX((PDAY2-PDAY1)/60)+1
22580 CCSECT = 0
22590 LPRINT CHR$(12)
22600 FOR PPP = PDAY1 TO PDAY2
22610     SKIP = SKIP + 1 : IF SKIP < 61 GOTO 22720
22620     CCSECT = CCSECT + 1
22630     FLEAT = PPP : GOSUB 22800 : GOSUB 22990
22640     PRINT : LPRINT
22650     PRINT USING "####";FLEAT; : PRINT "--";PDAY$;TAB(17);MONTH$;" "; : PRINT USING "##";DAY; : PRINT ", "; : PRINT USING "####";YR;
22660     LPRINT USING "####";FLEAT; : LPRINT "--";PDAY$;TAB(17);MONTH$;" "; : LPRINT USING "##";DAY; : LPRINT ", "; : LPRINT USING "####";YR;
22670     IF SECT > 1 THEN FLEAT = PPP + 60
22680     IF SECT > 1 AND FLEAT <= PDAY2 THEN : GOSUB 22800 : GOSUB 22990 : GOSUB 22750
22690     IF SECT > 1 AND CCSECT > 59 THEN SKIP = 0
22700     IF CCSECT > 59 THEN CCSECT = 0 : SECT = SECT - 1 : LPRINT CHR$(12)
22710     IF SECT <= 0 THEN PRINT : LPRINT : GOTO 10350
22720 NEXT PPP
22730 LPRINT : PRINT
22740 GOTO 10350
22750 REM
22760 REM **** SUBROUTINE FOR SECOND COLUMN PRINT ****
22770 PRINT TAB(40); : PRINT USING "####";FLEAT; : PRINT "--";PDAY$;TAB(56);MONTH$;" "; : PRINT USING "##";DAY; : PRINT ", "; : PRINT USING "####";YR;
22780 LPRINT TAB(40); : LPRINT USING "####";FLEAT; : LPRINT "--";PDAY$;TAB(56);MONTH$;" "; : LPRINT USING "##";DAY; : LPRINT ", "; : LPRINT USING "####";YR;
22790 RETURN
22800 REM
22810 REM **** SUBROUTINE TO SET YEAR, MONTH  AND DAY ****
22820 Y(1)=365   Y(2)=365   Y(3)=365 : Y(4)=366
22830 YR=1981   DAY=FLEAT
22840 WHILE DAY > 1461
22850    DAY = DAY-1461
22860    YR = YR + 4
22870 WEND
22880 FOR N = 1 TO 4
22890     IF DAY > Y(N) THEN YR = YR + 1 : DAY = DAY - Y(N)
```

```
22900 NEXT N
22910 IF YR/4 <> FIX(YR/4) THEN LEAPDAY = 0 ELSE LEAPDAY = 1
22920 M(1)=31 : M(2)=28+LEAPDAY : M(3)=31 : M(4)=30 : M(5)=31 : M(6)=30 : M(7)=31 : M(8)=31 : M(9)=30 : M(10)=31 : M(11)=30 : M(12)=31
22930 MO = 1
22940 FOR N = 1 TO 12
22950     IF DAY > M(N) THEN DAY = DAY - M(N) ELSE RETURN
22960     MO = MO + 1
22970 NEXT N
22980 RETURN
22990 REM
23000 REM **** SUBROUTINE TO SET THE DAY OF WEEK AND MONTH ****
23010 PDAY$ = "WEDNESDAY" : ZEE = FLOAT/7-FIX(FLOAT/7)
23020 IF ZEE > 9.999999E-06 THEN PDAY$ = "THURSDAY"
23030 IF ZEE > .142860 THEN PDAY$ = "FRIDAY"
23040 IF ZEE > .285725 THEN PDAY$ = "SATURDAY"
23050 IF ZEE > .428583 THEN PDAY$ = "SUNDAY"
23060 IF ZEE > .571439 THEN PDAY$ = "MONDAY"
23070 IF ZEE > .714296 THEN PDAY$ = "TUESDAY"
23080 MONTH$ = "JANUARY"
23090 IF MO > 1 THEN MONTH$ = "FEBRUARY"
23100 IF MO > 2 THEN MONTH$ = "MARCH"
23110 IF MO > 3 THEN MONTH$ = "APRIL"
23120 IF MO > 4 THEN MONTH$ = "MAY"
23130 IF MO > 5 THEN MONTH$ = "JUNE"
23140 IF MO > 6 THEN MONTH$ = "JULY"
23150 IF MO > 7 THEN MONTH$ = "AUGUST"
23160 IF MO > 8 THEN MONTH$ = "SEPTEMBER"
23170 IF MO > 9 THEN MONTH$ = "OCTOBER"
23180 IF MO > 10 THEN MONTH$ = "NOVEMBER"
23190 IF MO > 11 THEN MONTH$ = "DECEMBER"
23200 RETURN
```

DISTRIBUTION

No. of Copies

OFFSITE

30 DOE Technical Information Center

K. J. Fischer
Department of Mechanical Engineering
Oregon State University
Corvallis, OR 97331

Dr. J. Welty
Department of Mechanical Engineering
Oregon State University
Corvallis, OR 97331

Mr. Phil Wendell
BPA - KES
P.O. Box 3621
Portland, OR 97208

Mr. Michael Warwick
BPA - KES
P.O. Box 3621
Portland, OR 97208

No. of Copies

ONSITE

DOE Richland Operations Office

K. W. Bracken

23 Pacific Northwest Laboratory

S. G. Hauser
P. C. Hays
M. S. Klan
R. P. Mazzucchi
G. B. Parker
W. F. Sandusky (5)
G. J. Schuster
T. J. Secrest
D. R. Simpson
G. M. Stokes
R. A. Stokes
J. J. Thomas
Publishing Coordination (2)
Technical Information (5)