ORNL/CON-59

ornl

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

Data Collection and Processing for the ACES

J. L. Bledsoe
D. R. Miller

Conservation
Office of Buildings
and
Community Systems

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A09 Microfiche A01

ORNL/CON-59

Contract No. W-7405-eng-26

ENERGY DIVISION

DATA COLLECTION AND PROCESSING FOR THE ACES

J. L. Bledsoe
Computer Sciences Division

D. R. Miller
Instrumentation and Controls Division

Department of Energy
Office of Buildings and Community Systems

Date Published: August 1981

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

RELATED REPORTS

H. C. Fischer et al., *Summary of Annual Cycle Energy System, Workshop I, Held October 29-30, 1975, at Oak Ridge, Tennessee,* ORNL/TM-5243 (July 1976).

J. C. Moyers et al., *Design Report for the ACES Demonstration House,* ORNL/CON-1 (October 1976).

E. A. Nephew et al., *The Annual Cycle Energy System: Initial Investigations,* ORNL/TM-5525 (October 1976).

I. G. Cantor, *Investigation of Alternate Types of Water Storage Components to be Utilized in Conjunction with the ACES System,* ORNL/Sub-78/14233/1 (December 1977).

E. C. Hise, *Performance Report for the ACES Demonstration House August 1976 through August 1977,* ORNL/CON-19 (March 1978).

H. C. Fischer, *Development and Testing of a Single-Plate and a Two-Plate Ice-Maker Heat Pump,* ORNL/CON-21 (April 1978).

A. S. Holman and V. R. Brantley, *ACES Demonstration: Construction, Startup, and Performance Report,* ORNL/CON-26 (October 1978).

V. D. Baxter, *Intermediate Report on the Performance of Plate-Type Ice-Maker Heat Pumps,* ORNL/CON-23 (October 1978).

A. S. Holman et al., *Annual Cycle Energy System (ACES) Performance Report November 1977 through September 1978,* ORNL/CON-42 (May 1980).

M. L. Ballou et al., *MAD: A Computer Program for ACES Design Using Monthly Thermal Loads,* ORNL/CON-51 (March 1981).

R. E. Minturn et al., *ACES 1979: Capabilities and Potential,* ORNL/CON-48 (June 1980).

L. A. Abbatiello et al., *Evaluations of ACES Residential Designs, Performance and Economics for 115 U.S. Cities,* ORNL/CON-52 (March 1981).

Honeywell Inc., Energy Resources Center, *Economic Analysis of Annual Cycle Energy Systems,* ORNL/Sub-7470/1-V1, V2, V3, Minneapolis, MN (May 1980).

E. A. Nephew, L. A. Abbatiello, and M. L. Ballou, *Theory and Design of an Annual Cycle Energy System (ACES),* ORNL/CON-43 (May 1980).

V. D. Baxter, *Ice-Maker Heat Pump Development: Final Report,* ORNL/CON-50 (September 1980).

V. D. Baxter, *ACES: Final Performance Report December 1, 1978, Through September 15, 1980,* ORNL/CON-64 (April 1981).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DATA COLLECTION AND PROCESSING FOR THE ACES

J. L. Bledsoe*
D. R. Miller†

ABSTRACT

The Annual Cycle Energy System demonstration house furnishes information that is collected, processed, and analyzed on a weekly schedule. The computer codes used for processing and analyses were designed to display collected data; to summarize the performance (mechanical) of the house for each week; to give representation of external influences such as temperature, humidity ratio, and wind speed; and to aid in the dissemination of data to other users. The computer codes used to perform these functions were written by A. S. Holman, formerly of the Conservation Technology Program. Revisions and adjustments were made to the codes to accommodate improvements made at the demonstration facility. The codes are written in either FORTRAN IV or PL/I programming languages. All programs in the system run on the IBM 360 systems.

1. INTRODUCTION

The concept of an Annual Cycle Energy System (ACES) was first presented to Oak Ridge National Laboratory (ORNL) by Harry C. Fischer in 1974. In an ACES, the basic idea is that whenever the heat pump is operated, both heating and cooling are performed. This idea differs from that of other heat pump applications.[1] In the ACES approach, cooling energy is stored for later use whenever heating is performed. As heat is extracted from the thermal energy storage bin during the heating season, the water in the bin is converted into ice. This stored ice provides the air-conditioning for the coming summer.

The technology at work in an ACES has been proven to meet the demand for electrically based heating and cooling systems and at substantially higher efficiencies than can alternate systems. The ACES discussed in

*Computer Sciences Division.
†Instrumentation and Controls Division.

this report has supplied the space heating, air-conditioning, and domestic water heating requirements of this demonstration for the past three years. This house is one of three constructed at the Technology for Energy Conservation in Housing (TECH) site near Knoxville, Tennessee. A second similar house, the control house, has been used for performance comparison of the ACES to both an electric resistance heating and hot water with central air-conditioning system and two air-to-air heat pump systems.

The third house, the solar house, is used by the University of Tennessee (UT) staff for experimental work and performance monitoring on the solar components. The data acquisition system (DAS) contained in the ACES house also scans input from the other two houses. Recently, the UT group placed solar components with pebble-bed heat storage in a garage near the solar house. This fourth "house" also has sensors that are scanned by the DAS.

Energy measurements in the ACES house can be separated into two types: (1) system input and (2) system output. Energy input is the electrical energy that the mechanical package consumes. Energy output is the energy delivered by the system in the form of space heating, space cooling, or domestic water heating. The first year's operation (November 1977 through mid-September 1978) indicated that the ACES consumed 9012 kWh of electricity while delivering an annual coefficient of performance (COP)* of 2.78. For the same period, the control house with electric resistance heating and hot water consumed 20,523 kWh and delivered an annual COP of 1.13.[2]

After this test cycle ended in September 1978, additional insulation was added to the ACES storage bin and a new ice-bin coil system was installed. Changes in the DAS were also implemented during this period. During the first test year, the ACES data file contained 80 variables. The upgrade of 1978 increased the file size to 160 variables. New sensors were added to take advantage of the increased capability, and the DAS was altered to conform. All data processing and reduction routines were likewise modified to reflect the effects of the upgrade.

---

*The annual COP is equivalent to the annual energy delivered per annual energy purchased.

The second annual test cycle began December 1, 1978. During this period, the ACES was compared with an air-to-air heat pump. From December 1, 1978, to September 30, 1979, the ACES consumed 6719 kWh of electricity while the control house used 12,853 kWh. The COPs were 2.80 for ACES and 1.42 for the control house.

The third annual test cycle ran from November 30, 1979, to September 15, 1980. At the termination of this test cycle, the ACES had consumed 6467 kWh of electricity while the control house had used 11,393 kWh. Annual COPs were 3.08 and 1.73 for ACES and the control house, respectively.

During test cycles, the DAS monitored and collected data in a reliable manner. Downtime for both test cycles remained so small that reliability is in the range of 95-100%.

2. SUMMARY

The ACES program is an ongoing research, development, and design effort of ORNL sponsored by the U.S. Department of Energy (DOE). Primarily, the objectives of the ACES project have been to study, develop, and evaluate the engineering performance and commercial viability of such systems. Data collection and processing efforts have provided the information that enabled project staff members to fulfill these obligations. The data were collected at the ACES demonstration house located near Knoxville, Tennessee.

This demonstration house has been used continuously in studying the operational characteristics of the ACES under residential conditions. Operational study has furnished considerable information about the ACES system and component performance. Utilization of this information has led to improved performance and reliability of the system.

Documentation of the data collection and processing for the ACES is the overall purpose of this report. For those wishing to use the compiled ACES operational data, this report furnishes necessary guides and definitions.

Data monitoring equipment in the ACES house consists of sensors, data scanners, a Hewlett-Packard 9825 calculator/controller, a Tennessee Valley Authority (TVA) weather box, counters, the instrument interface box, an uninterruptible power supply, and a digital voltmeter (DVM). Signals from the sensors are detected by the DVM and the counters. The programmable calculator is used to scan these signals, convert them to the proper units, and store them on cassette tape.

A weekly cassette tape copy is made of all the monitored data. When the cassette data are processed on the ORNL computers, a copy of the raw data is saved, along with the version that is filtered, corrected, and stored on disks.

ACES demonstration data analysis begins with the program FILER, in which specific input values are used to calculate the ACES equipment power consumption, house loads, solar radiation, outside air specific volume, and the economy-cycle cooling load. Further analyses are made on the data when it is processed by the SUMMARY program.

Within SUMMARY, a least squares fit of the building loads vs outdoor temperature is made on the weekly data file. A time-of-day power consumption graph for both ACES and the control house is also made for the 24-h period with the highest electrical energy consumption.

Several summaries of the input weekly data are made routinely. The weekly computer summaries for building loads and consumptions are maintained along with a continuous (year-to-date) summary. Monthly summaries are made on the composite-year file, which is generated from the most recent 52 weeks of data. The program that creates these summaries also calculates the monthly degree days, solar radiation, average temperature, and hours of occurrence for 5°F temperature ranges (bins).

3. DATA ACQUISITION SYSTEM (DAS)

3.1 Data Collection at the Demonstration House

The DAS is located in the ACES house and automatically monitors all three houses in the TECH complex as well as the new solar-assisted heat pump experiment and the weather data. The DAS consists of sensors measuring physical parameters, interface and conditioning circuits, and a programmable calculator for control and analysis.

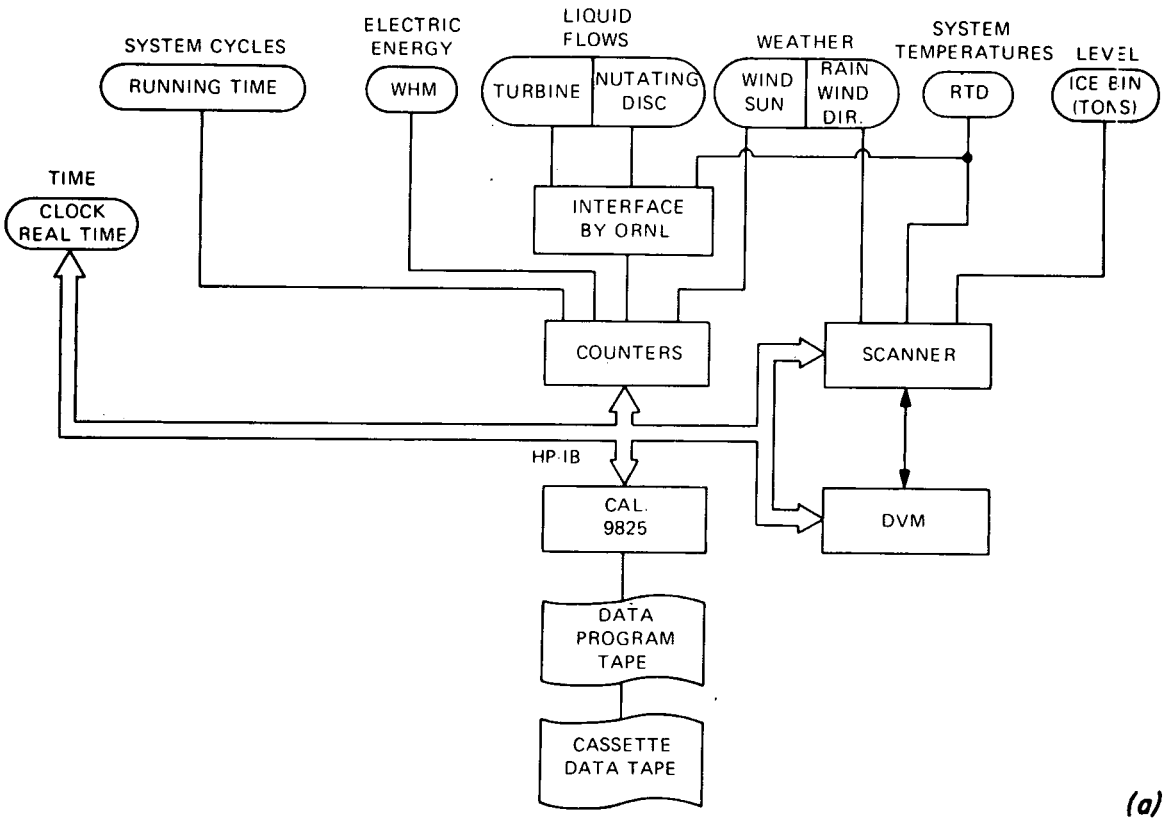
Two types of data are being recorded at the TECH complex: analog signals, such as temperatures, and digital signals such as integrated heat flows and electrical consumption. The analog signals, sequentially selected at preprogrammed intervals, are read by a DVM, which is read by the programmable calculator, and the raw data are converted to scaled engineering units. The digital signals are conditioned and/or multiplied in the external electronic interface, and the result is integrated on digital counter cards, which are read by the computer; the signals are then converted to scaled engineering units. Typically these data are read hourly. The analog data are the instantaneous values at the time of scan, and the digital values are the sums for the hour. After the scan, the digital values are reset to zero, and the next hour's data begins to accumulate. Figure 1 is a schematic of the data acquisition operation.

The operation of the DAS can be described in detail for those measurements involved in COP calculations. These measurements are electrical energy inputs, temperature differentials, and flows.

To measure the performance of the ACES heat pump, three types of sensors are used. These are watt-hour meters, resistance temperature devices (RTDs), and liquid flowmeters.

Energy input to the system is measured through the use of several standard utility company watt-hour meters. These devices indicate the total electrical energy supplied to the individual loads from the single outside revenue meter. The arrangement of dials in the watt-hour meter is simply a revolution counter for recording the total angular displacement of the disk. A device has been developed and added to each of the

ORNL-DWG 80-20589



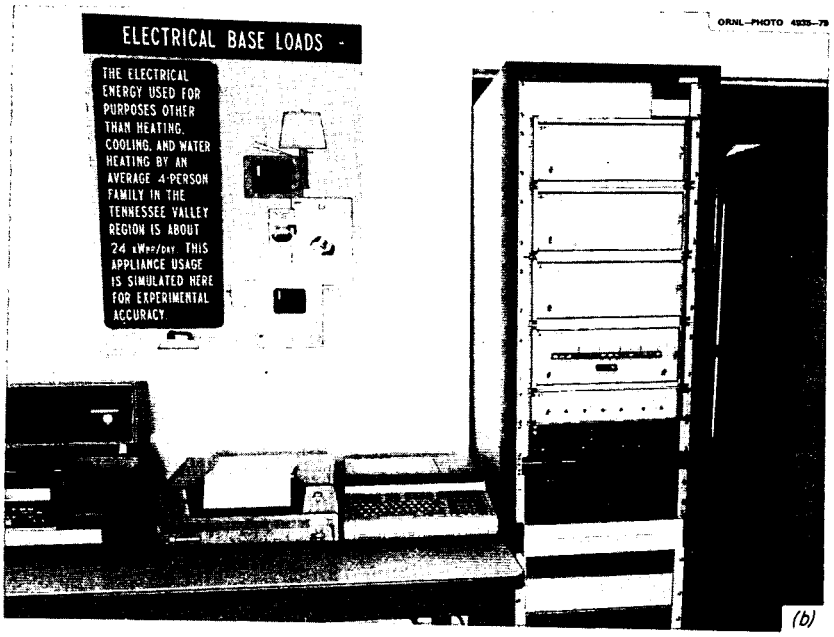


Fig. 1. Data acquisition system: (a) schematic (note: WHM = watt-hour meter, RTD = resistance temperature device, and DVM = digital voltmeter); (b) photograph.

ACES watt-hour meters which transmits this angular displacement to the data system in the form of pulses. It is entirely nonmagnetic and offers no mechanical friction to the standard meter. This device "reads" the rotation of the disk by emitting an infrared light beam that is reflected by the disk. An infrared solid state sensor detects the reflected light and transmits an electrical logic signal to appropriate conditioning electronics. With the addition of small black spots, which periodically interrupt the reflected signal, one can resolve almost any fraction of a disk rotation. During the time the black or nonreflective area is within the near field of the light source, no signal is transmitted. This logic signal, which is alternately "high" and "low" during disk rotation, is conditioned to generate a constant voltage level and duration regardless of disk speed. This ensures that the system counter accurately records the electrical usage. A typical conditioning electronics package contains circuits for up to eight individual meters.

Temperatures in the various heat exchanger inlet and outlet lines are measured by platinum resistance thermometers; resistance of the platinum element is directly proportional to the temperature. A high-stability, constant-current source has been developed. Constant-current sources operate by buffering a stable reference voltage through a current drive circuit. When applied to the platinum element, the constant-current sources generate a voltage proportional to the element's resistance and, thus, its temperature. These constant-current sources have been packaged on printed circuit cards with four sources per card. The cards for the temperature sensors in all three houses are located in an electronic chassis or interface.

Liquid flow in heat exchanger lines is measured with two types of flowmeters. The preferred and most accurate method is an in-line turbine meter with variable reluctance-type pick-up of the mechanical motion of the turbine blades. This method is used on all but the hot water loop flow measurement. The electrical signal of the turbine meter is approximately 25-30 millivolts and thus must be amplified electronically. This is done by a specially developed circuit that is located in the interface electronics. The millivolt signal is amplified approximately 400 times

and digitally shaped to effect an accurate flow value by subsequent circuits. A nutating disk-type positive displacement meter is used in the hot water loop flow due to the extremely wide range of normal operation. This meter produces 10-V pulses via a magnetically operated relay and can be used directly in subsequent circuits. The nutating disk meter is capable of indicating a flow of very low value; however, it exhibits a larger error than the turbine type.

The interface electronics are an integral part of the DAS operation. These contain power supplies, constant current supplies, flowmeter signal conditioning, a unique differential temperature-to-frequency module, predivide logic, and gating logic. Gating logic selects the correct counter for heat accumulation or integration over the hour.

The basis for the temperature-to-frequency module is that heat Q as defined by the equation

$$Q = mc(T - t) ,$$

where

m = mass flow,
c = specific heat capacity,
T = inlet temperature of the heat exchanger,
t = outlet temperature of the heat exchanger,

is directly proportional to the difference between an inlet and an outlet temperature of a heat exchanger. It can be seen that the value of m is known from the flowmeter electrical output. The value for c is a constant which is part of the software in the calculator.

A differential temperature is available by algebraic addition of the two electrical signals from the heat exchanger sensors.[3] This electrical signal is connected to the input of a sophisticated instrumentation amplifier manufactured by Analog Devices. The resultant voltage is tested for polarity to evaluate the direction of heat flow and converted to a frequency by another integrated circuit which has a linear relationship between voltage input and frequency output.

A raw heat flow signal is the result of using the flowmeter pulses to "gate" temperature pulses through a standard logic "AND" gate circuit. This raw pulse chain is divided by a standard constant of 6144 at this point to ensure that the counter will not overflow during a normal hour's accumulation period.

This raw, predivided pulse chain must be counted or integrated in the proper counter. The temperature-to-frequency module supplies a control logic signal based on the polarity test it performs which directs the raw pulse chain to the correct DAS counter. Actual heat flows in the heat exchanger can be accurately calculated from stored electrical data with the calibration of appropriate constants based on field measurements.

The weather sensors and interface electronics package were supplied by TVA. They consist of barometric pressure sensor/transmitter 3-cup anemometer, wind vane, motor aspirated temperature shield with temperature probe and lithium chloride dewcell probe, storage-type rain gauge with potentiometric transmitter, and Eppley black and white pyranometer for global radiation (total sun and sky).

3.2 Data Processing at ORNL

After data is collected, the week's data is copied onto a cassette tape and from there into storage on disk. Information on the cassette tape is placed on the Digital Equipment Corporation (DEC) PDP-10 computer by reading the cassette file in a manner similar to that of a paper tape. The cassette is mounted on the reader of a Texas Instrument 700 computer terminal. The individual transmitting the data accesses the DEC PDP-10 either by dialing up or using the hardwired features.

When the PDP-10 is available, the cassette is physically positioned. Peripheral Interchange Program instructions are given to create a weekly file on the PDP-10 disk with a corresponding Julian date name. For example, a file that begins on Monday, March 17, 1980, has the name "C80077.DAT". The "C" refers to cassette, "80" is the year, and "77" is the Julian date of the Monday of the week. Except for special cases, all weekly files begin on Mondays.

After the file is completely read, the Job Control Language (JCL) for program PUNCH is edited for the current week's file. This routine writes the new data onto a temporary disk storage (SPDA). Listing of each week's data file by the hour enables the user to do quick editing of the output file using TECO or EDITS,[4] DEC-system editors. Occasional transmission errors result in failure to transfer some of the data to the PDP-10. This missing data must be reinserted later. Using the hardwired lines usually alleviates most transmission problems.

Upon completion of editing on the PDP-10, a copy of the raw data is stored on a DECtape. (See Appendix E for a listing of these data.) After several months pass, these DECtape files may be archived to a 9-track tape as a further safeguard against loss of information.

The data stored on SPDA disk is processed by program FILER. In this step, the Julian date cassette data files for two consecutive weeks are used as input to the program. For example, "C80070" and "C80077" would be used to represent data for the week of March 17, 1980. The output from this step is a disk file containing 168 hours with 160 variable readings per hour. A copy of this disk file is written on the ACES data disk, a 2314 IBM disk, and on the semipermanent disk, a 3330 disk. The SPDA disk files are accessible from TSO (Timesharing Option) and batch. This copy of the data would be named "D80077", where "D" refers to disk and "80077" has the same meaning as stated above. On the semipermanent disk the full name of the file would be "T.IJL0936.D80077", and on the private disk, the full name would be "ACES.DATA.D80077".

These disk files are used as input when tape copies are made. A part of the ACES research responsibilities is supplying data to others, for example, TVA and UT. Magnetic tapes containing the data are sent to these facilities. The program UTTVA is used to copy all 160 variables for each hour of the 168 hours in the file. The tapes are sent to the UT experimenters and TVA Solar Heating Section. These users return the tapes upon copying the complete files.

3.3 Data Description

The final dataset produced from the weekly monitoring has a total of 160 readings per hour for a 7-d week. Some of the 160 readings are null because these are positions that are not in use. (See Appendix D for detailed listings of these variables.) Prior to the fall 1978 upgrade, there were only 80 readings per hour. (See Table 1 for 80- variable listing.) Table 2 contains a listing of the 160 variables available after the upgrade.

3.4 Data Reduction

In the process of reading and testing the weekly data within the filing program, if the values of records fail to fall within the "high" and "low" limits of data intervals, these values are given the default value "99999." Also, if certain hours of the weekly file were missing, all values for those occurring within the missing hours are given the default value.

Some data that is input is used to calculate certain other specific data that is not input. These new data that exist on the disk file occupy some of the null positions of the original file (see Table 2).

Retrieving data stored on private disk is possible through the use of the COPY program. This routine will copy any number of records for one day or an entire week for any file stored on semipermanent disk (SPDA) or in permanent disk storage.

3.5 Data Analysis

ACES demonstration data analysis begins in the program FILER where specific input values are used to calculate the ACES electrical consumption, solar radiation on 30 deg inclined and horizontal surfaces, outside air specific volume, and the economy-cycle cooling load.

Total ACES consumption is calculated in FILER as follows:

ATOT = total ACES consumption = variable 141 ,

ATOT = [(PCLG) + (PHTG) + (PHWT) + (COMP) + (BLWR)] ,

Table 1. Variables of ACES dataset before 1978 upgrade[a]

Variable number	Variable meaning
1	ACES ice storage bin level (kg)
2	ACES ice storage bin temperature low (C)
3	ACES inside dry bulb temperature (C)
5	ACES ice storage bin coil temperature t1 (C)
6	ACES ice storage bin coil temperature t2 (C)
7	ACES fan coil temperature t1 (C)
8	ACES fan coil temperature t2 (C)
9	ACES solar panel temperature t1 (C)
10	ACES solar panel temperature t2 (C)
11	ACES domestic hot water temperature t1 (C)
12	ACES domestic hot water temperature t2 (C)
13	Bin temperature high (C)
14	Control house dry bulb temperature (C)
15	ACES dry bulb temperature at solar panel (C)
16	Control house humidity ratio (kg_w/kg_a)[b]
18	Solar pump 1 inlet temperature t4 (C)
19	Solar collector temperature t1 (C)
20	Solar collector temperature t2 (C)
21	Solar hot water storage tank temperature top (C)
22	Solar hot water storage tank temperature (C)
23	Solar hot water storage tank temperature (C)
24	Solar hot water storage tank temperature bottom (C)
25	Solar collector tank 1 high (C)
26	Solar collector tank 1 low (C)
27	Solar heating coil loop temperature t7 (C)
28	Solar heating coil loop temperature t8 (C)
29	ACES inside humidity ratio (kg_w/kg_a)
30	Solar collector tank 2 high (C)
31	Solar collector tank 2 low (C)
32	Solar inside humidity ratio (kg_w/kg_a)
33	Solar inside dry bulb temperature (C)
34	Outside dry bulb temperature (C)
35	Outside humidity ratio (kg_w/kg_a)
36	Outside barometric pressure (mm of Hg)
37	Outside air specific volume ($m*m*m/kg_a$)
38	Outside wind direction (deg)
39	Solar radiation horizontal (Wh/m/m)
40	Rainfall (mm)
41	ACES ice storage bin heat input (Wh)
42	ACES ice storage bin heat output (Wh)
43	ACES solar panel heat input (Wh)
44	ACES solar panel heat output (Wh)
45	ACES cooling pump watt-hour meter (Wh)
46	ACES heating pump watt-hour meter (Wh)
47	ACES domestic hot water watt-hour meter (Wh)
48	ACES compressor watt-hour meter (Wh)
49	ACES fan watt-hour meter (Wh)
50	ACES house total watt-hour meter (Wh)

Table 1 (continued)

Variable number	Variable meaning
51	ACES fan coil output (Wh)
52	ACES fan coil input (Wh)
53	ACES domestic hot water output (Wh)
54	Control house watt-hour meter (Wh)
55	ACES total system kWh (Wh)
56	Solar collector heat flow (Wh)
57	Solar hot water heater output (Wh)
58	Solar hot water heater output (Wh)
59	Solar house total watt-hour meter (Wh)
60	Solar fan watt-hour meter (Wh)
61	Solar hot water heater watt-hour meter (Wh)
62	Solar collector pump watt-hour meter (Wh)
63	Solar space heater watt-hour meter (Wh)
64	Solar fan coil pump watt-hour meter (Wh)
65	ACES damper on time (s)
66	Solar radiation 45 deg (Wh/m/m)
67	Outside wind speed (km/h)
68	Control house hot water watt-hour meter (Wh)
70	Solar radiation 30 deg (Wh/m/m)
78	Economy-cycle cooling load (Wh)
79	Integration time for scan (s)
80	Time

[a]Omitted variable numbers are those for which no data is taken.

[b]Humidity ratio is usually expressed as $(\text{mass}_{\text{water}})/(\text{mass}_{\text{air}})$ in moist air.

Table 2. Variables of ACES dataset after 1978 upgrade[a]

Variable number	Variable meaning
1	ACES ice storage bin heat input from heat pump (Wh)
2	ACES ice storage bin heat output to heat pump (Wh)
3	ACES solar panel heat output (Wh)
4	ACES solar panel heat input (Wh)
5	ACES cooling pump watt-hour meter (Wh)
6	ACES heating pump watt-hour meter (Wh)
7	ACES domestic hot water watt-hour meter (Wh)
8	ACES compressor watt-hour meter (Wh)
9	ACES fan watt-hour meter (Wh)
10	ACES house total watt-hour meter (Wh)
11	ACES fan coil heat output (Wh)

Table 2 (continued)

Variable number	Variable meaning
12	ACES fan coil heat input (Wh)
13	ACES domestic hot water output (Wh)
14	ACES economy-cycle seconds dh
16	ACES economy-cycle seconds h27
19	Solar air handler watt-hour meter (Wh)
20	Heat pump indoor (Wh)
21	Heat pump outdoor (Wh)
22	Off peak heater (Wh)
23	Heat pump auxiliary heater (Wh)
24	Solar collector — space (Wh)
25	Heat pump — space (Wh)
26	Solar collector pebble bed (Wh)
27	Pebble bed — space (Wh)
28	Total power (Wh)
31	Solar collector heat flow (Wh)
32	Solar water to refrigerant coil heat flow (Wh)
33	Solar water to air coil heat flow (Wh)
34	Solar hot water heater output (Wh)
35	Solar house total watt-hour meter (Wh)
36	Solar blower watt-hour meter (Wh)
37	Solar hot water heater watt-hour meter (Wh)
38	Solar collector pump watt-hour meter (Wh)
39	Solar duct heater watt-hour meter (Wh)
40	Solar heating pump watt-hour meter (Wh)
41	Solar heat pump watt-hour meter (Wh)
42	Solar refrigerant-to-air coil heat flow (Wh)
46	Solar radiation 45 deg (Wh/m/m)
47	Outside wind speed (km/h)
48	Control house watt-hour meter (Wh)
49	Control house hot water heater watt-hour meter (Wh)
50	Control house indoor unit watt-hour meter (Wh)
51	Control house outdoor unit watt-hour meter (Wh)
52	Control house fan coil heat input (Wh)
53	Control house fan coil heat output (Wh)
61	ACES ice storage bin level (kg)
62	ACES ice storage bin temperature low (C)
63	ACES bin temperature high (C)
64	ACES ice storage bin coil inlet temperature (C)
65	ACES ice storage bin coil outlet temperature (C)
66	ACES fan coil temperature inlet (C)
67	ACES fan coil temperature outlet (C)
68	ACES solar panel temperature inlet(C)
69	ACES solar panel temperature outlet (C)
70	ACES domestic hot water temperature inlet (C)
71	ACES domestic hot water temperature outlet (C)

Table 2 (continued)

Variable number	Variable meaning
72	ACES inside dry bulb temperature (C)
73	ACES inside humidity ratio (kg_w/kg_a)[b]
80	Pebble bed 1 temperature (F)
81	Pebble bed 2 temperature (F)
82	Pebble bed 3 temperature (F)
83	Pebble bed 4 temperature (F)
84	Pebble bed 5 temperature (F)
85	Pebble bed 6 temperature (F)
86	Pebble bed 7 temperature (F)
87	Garage dry bulb temperature (F)
88	Garage wet bulb temperature (F)
96	Rainfall (mm)
97	Outside wind direction (deg)
98	Outside barometric pressure (mm of Hg)
99	Outside humidity ratio (kg_w/kg_a)
100	Outside dry bulb temperature (C)
102	Solar panel inlet temperature (C)
103	Solar panel outlet temperature (C)
104	Solar hot water storage tank temperature top (C)
105	Solar panel unpumped temperature (C)
106	Solar panel pumped temperature (C)
107	Solar hot water storage tank temperature bottom (C)
108	Solar collector tank 2 high (C)
109	Solar collector tank 2 low (C)
110	Solar collector tank 1 high (C)
111	Solar collector tank 1 low (C)
112	Solar collector unit temperature inlet (C)
113	Solar heating unit temperature outlet (C)
114	Solar inside dry bulb temperature (C)
115	Solar inside humidity ratio (kg_w/kg_a)
136	Control house fan coil inlet temperature (C)
137	Control house fan coil outlet temperature (C)
138	Control house dry bulb temperature (C)
139	Control house humidity ratio (kg_w/kg_a)
140	Control house crawl space temperature (C)
141	ACES total system kWh (Wh)
142	Solar radiation 30 deg (Wh/m/m)
143	Solar radiation horizontal (Wh/m/m)
144	Outside air specific volume (m*m*m/kg)
145	Economy-cycle cooling load (Wh)
158	Integration time for scan (s)
159	Month day
160	Hour minute

[a] Omitted variable numbers are those for which no data is taken.

[b] Humidity ratio is usually expressed as ($mass_{water}/mass_{air}$) in moist air.

where

PCLG = ACES cooling pump watt-hour meter = variable 5,
PHTG = ACES heating pump watt-hour meter = variable 6,
PHWT = ACES domestic hot water watt-hour meter = variable 7,
COMP = ACES compressor watt-hour meter = variable 8,
BLWR = ACES fan watt-hour meter = variable 9.

Solar radiation variables calculated in FILER are as follows:

$$\text{solar cloud cover} = \text{sqrt}\{[1. - (\text{QRAD})/(\text{DIRECT} + \text{DIFFUSE})]/0.65\} ,$$

where

QRAD = solar radiation from 45° surface = variable 46,
DIRECT = direct solar radiation for a 45° surface,
DIFFUSE = diffuse solar radiation for a 45° surface;

$$\text{solar radiation surface } 30° = (\text{DIRECT} + \text{DIFFUSE}) * (1. - 0.65 * \text{SCC}^2) ,$$

where

DIRECT = direct solar radiation on 30° surface,
DIFFUSE = diffuse solar radiation on 30° surface,
SCC = solar cloud cover;

$$\text{solar radiation horizontal} = (\text{DIRECT} + \text{DIFFUSE}) * (1. - 0.65 * \text{SCC}^2) ,$$

where

DIRECT = direct solar radiation for horizontal surface,
DIFFUSE = diffuse solar radiation for horizontal surface.

The economy cycle cooling load is derived below:

$$\text{economy-cycle cooling load} = [0.24 * (\text{HDRY}) + (\text{HWET}) * 0.444 * (\text{HDRY}) + 1061.] - [0.24 * (\text{TDRY}) + \text{WET} * 0.444 * (\text{TDRY}) + 1061.] * \text{ECSD} + \text{ECSH} * 1050./\text{VO} ,$$

where

HDRY = ACES inside dry bulb temperature = variable 72,
HWET = ACES inside humidity ratio = variable 73,
TDRY = outside dry bulb temperature = variable 100,
TWET = outside humidity ratio = variable 99,
ECSD = ACES economy-cycle seconds = variable 14,
ECSH = ACES economy-cycle seconds = variable 16,
VO = outside air specific volume = variable 144.

Further analyses are made on the data when it is processed by the SUMMARY program.

Seasonal heating and cooling COPs are calculated in SUMMARY and passed to PLOT. These coefficients are

$$COP_{heating} = (Q_h + Q_{hw})/Q_e \ ,$$

$$COP_{cooling} = (Q_c + Q_{hw})/Q_e \ .$$

Listed below are the calculations for the heating and cooling loads.

$$Q_h = (BIN_i - BIN_o) + (SLP_i - SLP_o) + Q_e - HWT_o \ ,$$

$$Q_c = FAN_i - BLWR - 0.5 * COMP - 0.75 * PUMPS \ ,$$

where

BIN_i = ACES ice storage heat input = variable 2,
BIN_o = ACES ice bin storage heat output = variable 1,
SLP_i = ACES solar panel heat input = variable 4,
SLP_o = ACES solar panel heat output = variable 3,
Q_e = ACES total system = variable 141,
HWT_o = ACES domestic hot water output = variable 13,
FAN_i = ACES fan coil input = variable 12,
BLWR = ACES fan power consumption = variable 9,
COMP = ACES compressor power consumption = variable 8,
PUMPS = ACES pump power consumption = variable 5 + variable 6.

A least squares fit of the building loads vs outside temperature is made on the weekly data file. The least squares fit in program SUMMARY is a first-order linear fit.[5] The regression model is

$$y(x) = c_1 + c_2 x = Q(\text{watt}) \ ,$$

where x, the independent variable, is outside temperature and y, the dependent variable, is building load. The coefficients c_1 and c_2 are chosen to minimize the sum of squares of the residuals — that is, to minimize R^2.

The residual at the ith data point is

$$R^2 = \Sigma \ [c_j o_j(t_i) - y_i i]^2 = \Sigma \ (y_i - \hat{y}_i)^2 \ ,$$

where

$\hat{y}_i$ = the estimated building loads,
y_i = the actual building loads.

In this particular model, c_1 is the y-intercept, c_2 is slope. Data for heating and cooling loads are "binned" according to the level of solar radiation for that specific hour of the day. These loads are grouped into "bins" for solar radiation less than 50 Btuh/ft^2, less than 150 Btuh/ft^2, and greater than 150 Btuh/ft^2. For each bin a regression equation is calculated. Also, a "break-even" temperature is calculated such that

$$\text{break-even} = -c_1/c_2 \ .$$

This is the outdoor temperature at which the internal loads will maintain the desired indoor temperature. For a given week there should be 168 data points for the hourly heating or cooling load. After binning the data, the sums for the fit are made. These are

Σ x, sum of outside air temperatures,

Σ y, sum of calculated building loads,

Σ xy, sum of product of temperature and load,

Σ x^2, sum of squares of temperatures.

The slope, c_2, and y-intercept, c_1, are calculated based on the above sums. So we have

$$c_1 = \text{slope} = \frac{\Sigma\, xy - \frac{\Sigma\, x\, \Sigma\, y}{n}}{\Sigma\, x^2 - \frac{\Sigma\, x\, \Sigma\, y}{n}},$$

$$c_2 = \text{y-intercept} = \frac{\Sigma\, y - c_2\, \Sigma\, x}{n},$$

where n is the number of data points.

It is important to note that, for the above calculations, SUMMARY does not include missing data as indicated by "99999." Thus, often the total number of points used in these fits is less than the number of hours in a week.

A time-of-day power consumption graph for both ACES and the control houses is also made for the 24-h period with the highest electrical energy consumption.

Several summaries of the input weekly data are made routinely. The weekly computer summaries for building loads and consumptions are maintained along with a continuous (year-to-date) summary. Monthly summaries are made on the composite-year data file. The program that creates these summaries also calculates the monthly degree days, solar radiation, average temperature, and hours for certain temperature ranges (bins).

4. SYSTEM OF COMPUTER CODES

4.1 System Makeup

The computer program system that processes and analyzes the ACES data has its starting point in the HP9825A calculator located at the demonstration site. The DAS program performs data conversions, stores the data, and gives diagnostics. The program for data transmission formats the data and transfers it to cassette tape.

After this cassette is read, a short routine, PUNCH, processes the data file. The cassette images are filed on a DECtape on the DEC-system and stored on a semipermanent disk area. Output from the PUNCH run is studied for apparent errors in the collected data. If there are no errors, the next step may be taken.

The program FILER is responsible for sorting data and calculating all loads "conversions." Data are scanned for "validity." Those values falling between the limits "high" and "low" are classified as "reasonable data." The output of this routine is the corrected dataset stored on semipermanent disk space.

The program SUMMARY uses the system dataset to produce input data arrays for the plotting program PLOT. A time-of-day power consumption printer plot displays the worst day for power usage. Least squares equations are calculated for the building loads vs the outside air temperature. Equipment and weather data are summarized in a chart that details energy collected, delivered, and consumed. Figure 2 shows a block diagram of the data processing steps.

In PLOT, the scratch disk area file passed from SUMMARY is used to generate a weekly set of three-color computer plots. The programs within this system are listed in Appendix A. A complete sample week that has been processed is included in Appendix E.

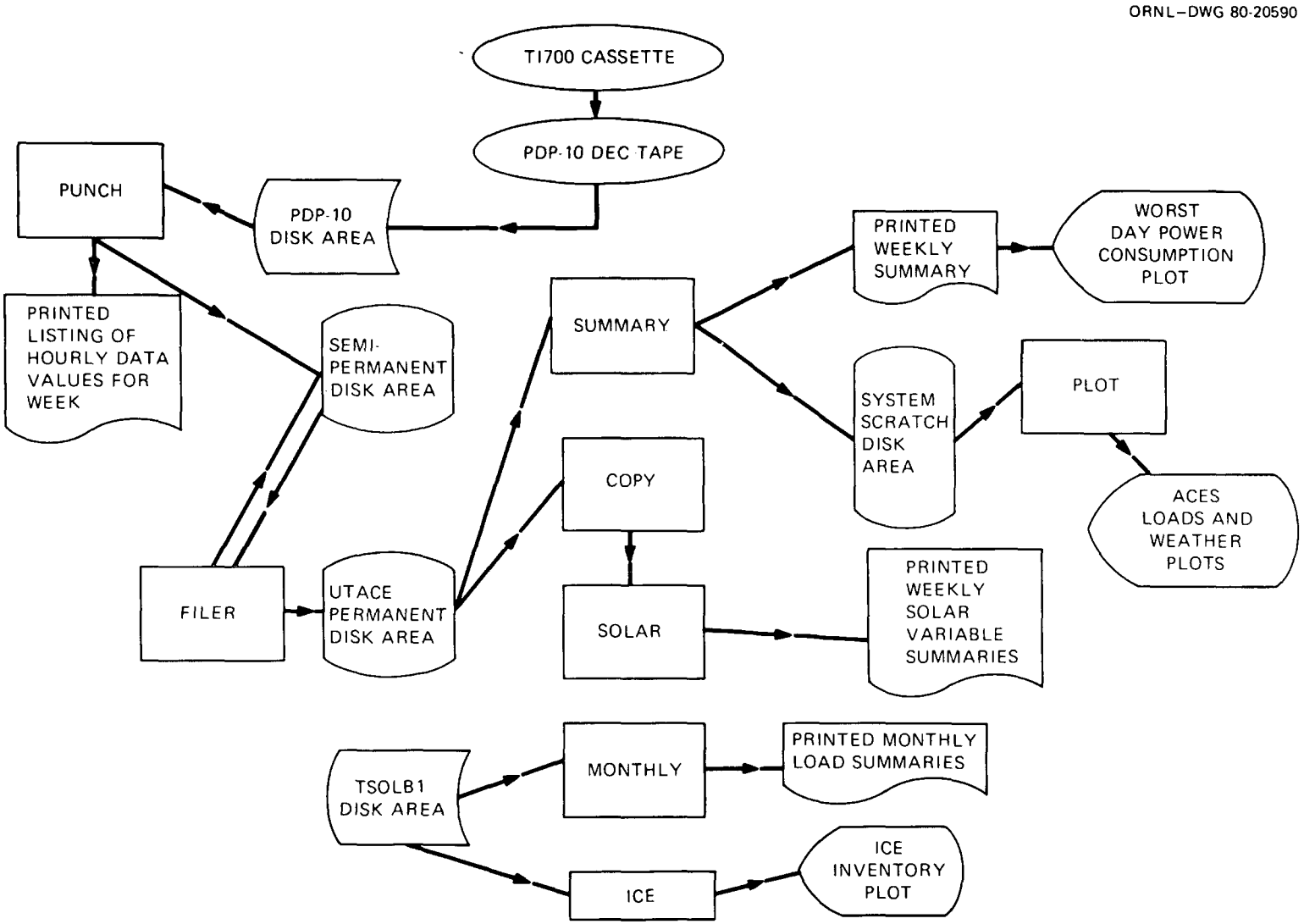


Fig. 2. Diagram for ACES data analysis and storage.

4.2 Computer Program Descriptions

4.2.1 PUNCH

This short routine functions as a data reader/writer. PUNCH reads the characters from the DEC-10 that were created from the weekly cassette of ACES data. After the file is read by PUNCH, it is printed for error-checking and is also output in a temporary file on disk.

A user will check the PUNCH output. If there are errors, the file must be edited and resubmitted. The new temporary data set on the disk will replace the previously created one.

This routine is written in standard FORTRAN and executes on the IBM 360 model machines.

4.2.2 FILER

This code consists of PL/I procedures that perform several functions. First of all the starting date for the data week is determined. The information for the starting date calculation is provided in the parameter field of the JCL EXEC statement for executing the load module for FILER. (See JCL listings.) After this determination is made, the starting date is placed at the beginning of the file for the dataset. Next, the input data is sorted based on the date-time group values as represented in the input data. This input consists of all data for the week. All input data is checked for missing values. Those found missing are flagged with "99999." values.

In addition to checking for missing values, FILER checks for reasonable data. Calculations are made of total ACES power consumption, solar cloud cover, solar radiation on the surface at 30° from vertical, solar radiation on a horizontal surface, and humidity ratio. Wet bulb temperatures and dew points are converted to humidity ratios. The sun's position, hour angle, and solar radiation are determined for use in the solar cloud cover and solar radiation calculations. Units are converted from metric to English units and the results reconverted to metric for storage.

The code compares the values of the above variables to appropriate reasonable data limits. If those values in the input are within their specified ranges, then the values are considered reasonable values. If data does not fall within those limits, it is replaced by the default value, "99999.".

Upon the completion of the sorting, checking, and calculations, the results and processed files are output to the printer and disk storage. The processed week has 160 variables. A flowchart for FILER is given in Fig. 3.

4.2.3 SUMMARY

SUMMARY is a multitask program[6] that produces printer output, prints missing data hours, and produces plot data arrays for a plotting program. Because these routines are written in PL/I, the "wait state" capability is used in the code execution. This allows all three of the main procedures in the code to execute simultaneously. The flow of logic is detailed in Fig. 4.

The first task, ANALIZE, returns the week number for the current data week, produces a least squares fit of the building loads vs the outside air temperature, and a time-of-day power consumption plot of ACES house vs the control house. In this segment of the program, the heating and cooling loads are calculated as indicated in Sect. 3.5.

The second task, MISSING, scans all values of variable 159 for the integration time. If this is "99999." in a particular hour, then there is no data for that hour. The records are then checked for missing data in variables not included in the scan check. At the conclusion of this task, all missing records are output to the printer.

PLOTPLI, the third task, reads the ACES data file and stores the data points in arrays. The primary function of this task is the creation of the plot file and the line printer information. This task allows for conversion of data values for humidity ratios and stores all building loads data into separate arrays for plotting. In addition, this task computes ACES system and compressor-only COPs.

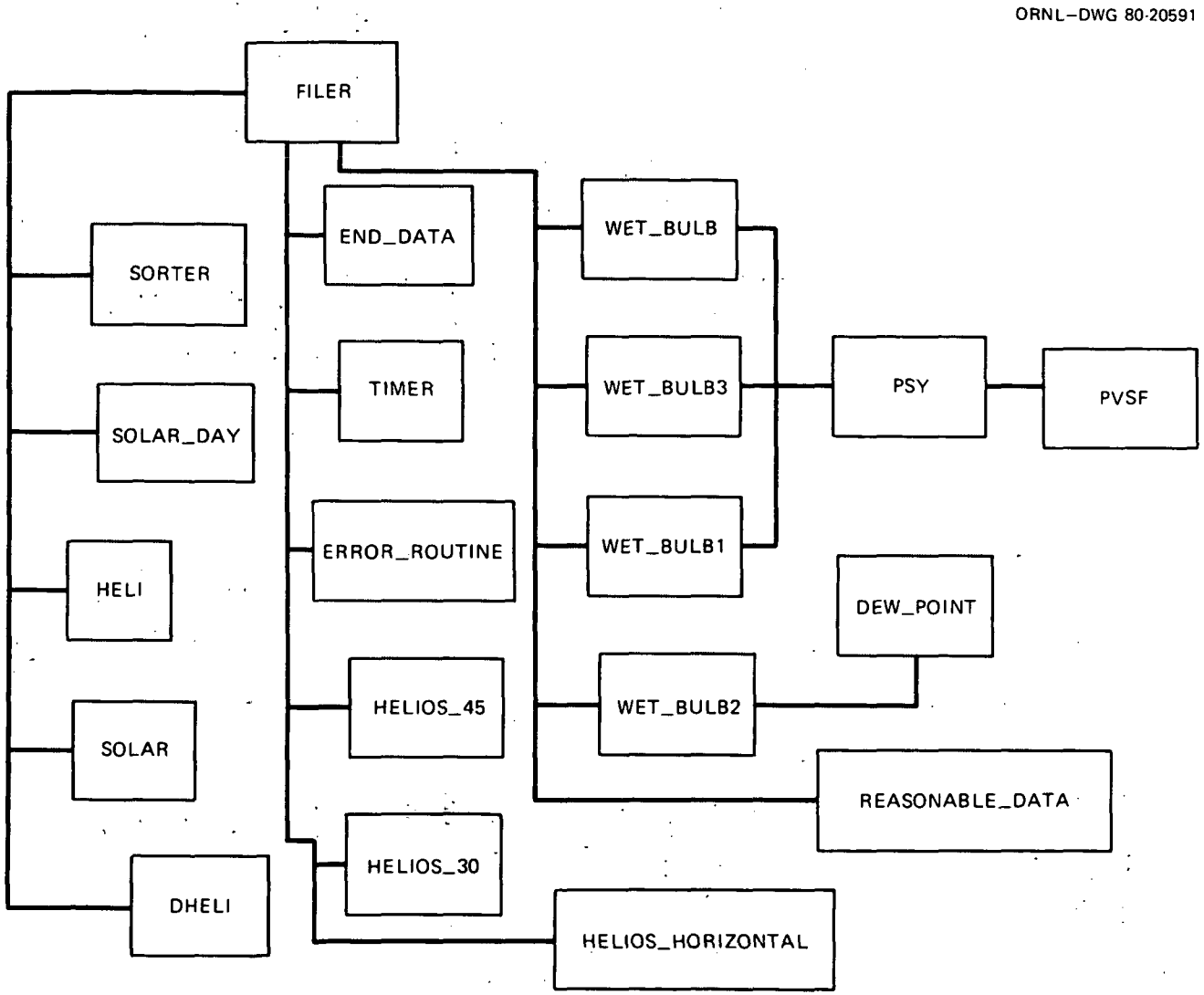


Fig. 3. The flow of logic in program FILER.

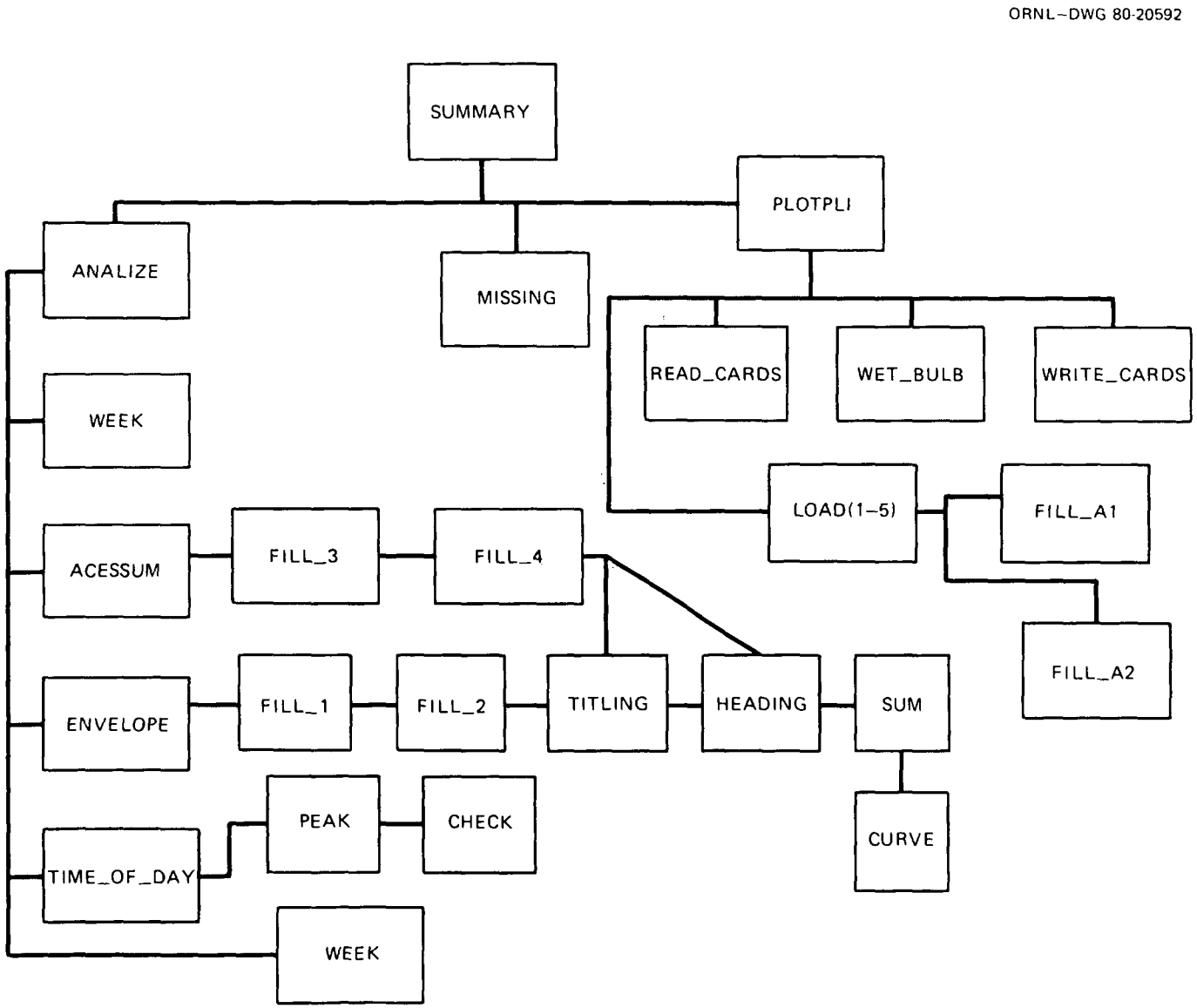


Fig. 4. The flow of logic in program SUMMARY.

4.2.4 MONTHLY

This code is used for updating the ACES data stored on the user's TSO region. It is an interactive program, written in standard FORTRAN, and executes on the IBM 360/370 through the use of a TSO command procedure.

The user supplies the code with the starting and ending months to be updated. For each month in the interval (starting month, ending month), the code will calculate heating and cooling loads in Btu. If an economy cycle is requested, the code will calculate enthalpy for temperatures less than 70°F. The code calculates degree days Fahrenheit. Another important feature in this code is the accounting for monthly hours within certain temperature ranges (bins). MONTHLY accumulates the hours of temperatures less than 10, 25, 35, 39, 65, and 80°F. Fan coil heat output, fan coil heat input, hot water, and solar radiation loads are calculated along with total monthly degree days.

After the monthly updates are made, these monthly totals are put together for a yearly update using the new monthly data. The calculations are output for each month and finally for the year.

4.2.5 SOLAR

This routine provides for the output of solar house data. SOLAR is written in standard FORTRAN. Summaries of solar data values are calculated from the extracted weekly data. These data extractions are performed by utilizing the COPY program. The code sums the data for solar collector heat flow, solar fan coil, space heater watt-hour meter readings, solar water heater, solar water heater watt-hour meter readings, solar fan watt-hour meter readings, solar collector pump watt-hour meter readings, and solar fan coil pump watt-hour meter readings. Total space heating is printed along with the hot water variables, total water energy, total solar system electricity consumption, and total heating and hot water electricity consumption.

4.2.6 PLOT

PLOT produces the report plots for the ACES data. When the SUMMARY code is executed in the command procedure, ACESFILE, this code is executed by an intermediate job step in the control language. SUMMARY's procedure PLOTPLI creates plotting arrays. The arrays are written onto the system scratch disk and passed to program PLOT. Nine plots are produced with multicolor ink, varying dashed line patterns, and legends through the use of DISSPLA software[7] and FORTRAN coding. A flowchart for PLOT is given in Fig. 5.

The first plot produced has five curves: outside air temperature (dry bulb and wet bulb), inside air temperature (dry bulb and wet bulb), and solar insolation. The second ACES data plot is the building loads plot with house heating, house cooling, water heating, economy cooling, and total house loads. Three curves on the third plot represent wind speed, solar insolation, and panel loads. ACES house power consumption, represented on the fourth plot, includes consumption for the pumps, compressor, fan, house equipment, and the total house power consumption. The fifth plot shows the COPs for the compressor and the system, and the total house load. The sixth plot produced takes the temperature data from the first plot and compares it to the control house loads. A further comparison of the ACES and control houses is made in plot seven. In this, daily power consumptions are compared. The eighth plot does a three-way comparison of the ACES, the control, and the solar houses for power consumption. And finally, the ninth plot is a power consumption plot for the control house. These five curves display the measurements of electrical consumption for the indoor and outdoor units, the water heater, the house equipment, and the house total watt-hour meter.

4.2.7 COPY

COPY is a utility routine written in the PL/I programming language. This program will copy records from the ACES data file and output these in the form of line printer output or punched cards. Execution of the routine occurs when the command procedure, ACESFILE, is executed interactively.

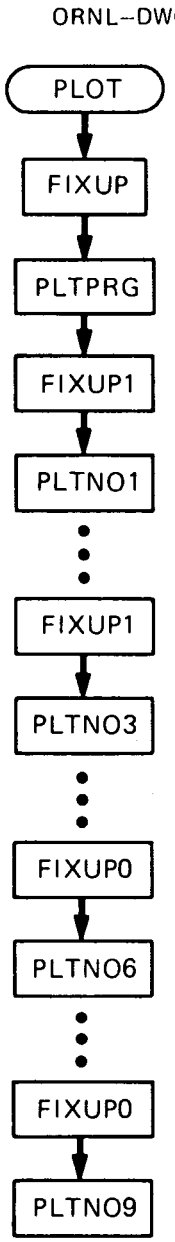


Fig. 5. The flow of logic in program PLOT.

This code has been used regularly in placing solar house data records in a data set that contains nine solar variables. Program SOLAR calculates summaries on this data and makes conversions from watts to kilowatts. These data are later plotted by the routine PLOT, for summary plots.

The user inputs to COPY those variables required and the number of days in the resultant data. Also, column or row output is specified.

4.2.8 ICE

This program, written in standard FORTRAN, calculates the amount of ice in the tank of the ACES facility. Also, upon inputting data of the six variables in the "yearfile," it will calculate the building heating and cooling loads. The yearfile contains the variables heating, cooling, hot water, outdoor air temperature, humidity ratio, and solar radiation.

After these loads and ice inventory are calculated for each day in the year, this code plots all of these quantities on a 30-in. plot. (See sample week.) The software used here is a combination of DISSPLA and FORTRAN.

4.2.9 UTTVA

UTTVA, a PL/I routine, creates the magnetic tapes that are sent to UT and TVA-Chattanooga. The program copies onto tape each variable in a processed weekly dataset. The resulting tape data set is written so that all values of the same variable occur on the tape in one group.

4.3 Command Procedures

Data updating functions are performed weekly for the ACES project. The use of TSO command procedures alleviate detailed editing and submitting of control files. Regularly used commands are incorporated into procedures that are executed weekly. (Appendix C has a listing of the command procedures.) (Appendix B lists the JCL that can be used to execute the codes if command procedures are not used.)

4.3.1 ACESFILE

When a new weekly data set is taken from its cassette tape and written onto the temporary disk, the data is ready to be reorganized using the FILER code.

The user will log onto TSO, first of all. Next, in order to execute the ACESFILE "procedure," the user will type:

```
EX CMDPROC(ACESFILE) "DSNAME(D79abc) NODSN(2)
DSN1(C79abc) — DSN2(C79abb) UT(TAGxyz) TVA(TAGxvw)"
```

This command string checks the name of the new dataset to be created. If the name is improper or missing, the procedure will type a short help file that explains the requirements of the procedure. Tagtape numbers are checked next. If these are missing, the user will be prompted to provide these.

When the first portion of this procedure is finished, a message is printed on the terminal telling the user that semipermanent and private disk files are being submitted. The second step in the procedure does a QED edit of the control file that submits the job. After editing, the control statements are submitted and the procedure compresses the data-set containing the job control files and signals the termination of the procedure.

ACESFILE procedure executes load modules for several routines: FILER, SUMMARY, COPY, PLOT, SOLAR, and UTTVA. IBM utility routines IEBPROGM, IEBGENER, and IEHLIST are used to scratch the data set name from the private disk, to copy a dataset to disk, and to list the names of the weekly datasets stored on the private disk.

4.3.2 YEARFILE

After ACESFILE has been executed, a copy of the disk dataset for the current week is stored on SPDA and the private disks (UTACE1 or UTACE2). The next step is updating the yearly data file with the six variables; fan coil heat output, hot water, fan coil heat input, outside dry bulb temperature, humidity ratio, and solar insolation.

First, the user will execute the command procedure that includes the newest week in the old data:

```
EX CMDPROC(YEARFILE) "D79abc"
```

where "79abc" is the Julian date for the Monday in the current week. This command will execute the program COPY which will copy those six variables for each hour of each day of the current week onto the end of the "IJL.ACESDATA.VERYNEW". Thus, the file, "IJL.ACESDATA.VERYNEW" is updated.

Correspondingly, the file stored on the TSO region known as "IJL.ACESDATA.VERYOLD" has to be updated by deleting the old week at the beginning of the file. The "yearfile" is written in 8F10.4 format. The first line of each "D79abc" file is a date written in the form MM/DD/YY. There are 126 lines since the file contains six variables and seven days and there are three lines per day of data. Using the large region editor, EDIT, the user deletes the top 127 lines, which are equivalent to the size of the week that was added to the "VERYNEW" file.

4.3.3 PROCYEAR

This command procedure uses the two updated files discussed above to create a processed resultant file, "IJL.ACESDEMO.YEARDATA". After the user has edited the "IJL.ACESDATA.VERYOLD", the "YEARFILE" is processed by: EX CMDPROC(PROYEAR). PROCYEAR concatenates the "VERYNEW" and "VERYOLD" files in this order (VERYNEW,VERYOLD). The file that results is 52 weeks long.

Processing this "ACESDEMO.YEARDATA" takes place immediately. An interactive routine TAPEFIX handles all processing on the six variables contained within the file. The user must respond to questions that the program will ask regarding values that do not fall within reasonable minimum and maximum values. This is a final step in reducing the data and eliminating inaccurate values.

4.3.4 MONTHLY

Monthly listings of the six variables in the "YEARDATA" file may be obtained by typing

```
EX CMDPROC(MONTHLY)
```

after the PROCYEAR command has been completed. MONTHLY will ask the user to choose the months of interest (starting and ending months).

The command procedure will execute FORTRAN program MONTHLY. The output will be printed in table format at the terminal.

REFERENCES

1. H. C. Fischer et al., *Summary of Annual Cycle Energy System, Workshop I, Held October 29-30, 1975, at Oak Ridge, Tennessee,* ORNL/TM-5243 (July 1976).

2. A. S. Holman et al., *Annual Cycle Energy System (ACES) Performance Report November 1977 through September 1978,* ORNL/CON-42 (May 1980).

3. The American Society of Heating, Refrigeration and Air Conditioning Engineers, Inc., *1977 Fundamentals,* New York, 1977.

4. Computer Systems Department, Oak Ridge National Laboratory's Computer Sciences Division, *ORNL Programmer's Notebook* (June 1979).

5. Norman Draper and Harry Smith, *Applied Regression Analysis*, Wiley, New York, 1966.

6. IBM Corporation, *IBM System/360 Operating System PL/I (F) Programmer's Guide,* San Jose, Calif., 1976.

7. Integrated Software Systems Corporation, *DISSPLA Manual,* Vols. 1 and 2, San Diego, Calif., 1970.

Appendix A

PROGRAM LISTINGS

Appendix A.1

PUNCH

```
 TY PUNCH.JCL
//IJLTS03 JOB (15936),'9102-1BLEDSOE',MSGCLASS=A
/*ROUTE  PRINT REMOTE50
//*CLASS CPU91=20S,IO=1.0
// EXEC SPDASCR
//SYSIN DD *
 T.IJL15936.C81075
//PUNCH EXEC FORTHCLG
//FORT.SYSPRINT DD SYSOUT=A
//FORT.SYSIN DD *
      DIMENSION CARD(20)
      K=0
 1000 WRITE(6,8000)
      DO 3000 J=1,2
      K=K+1
      DO 2000 I=1,14
      READ(5,4000,END=9000) CARD
      IF (I .EQ. 1) WRITE(6,6000) K
      WRITE(6,7000) I,CARD
      WRITE(10,4000) CARD
 2000 CONTINUE
      WRITE(6,5000)
 3000 CONTINUE
      WRITE(6,5000)
      GO TO 1000
 4000 FORMAT(20A4)
 5000 FORMAT(1H0,5X,100(1H*),/)
 6000 FORMAT(1H0,46X,'DATA FILE NO. ',I4,/5X,100(1H*),/)
 7000 FORMAT(9X,'CARD',I2,' --',20A4)
 8000 FORMAT(1H1)
 9000 WRITE(6,5000)
      STOP
      END
/*
//LKED.SYSIN DD *
/*
//GO.FT10F001 DD DSN=T.IJL15936.C81075,DISP=(NEW,CATLG),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=7280),UNIT=SPDA,
// SPACE=(TRK,(20,20),RLSE)
//GO.FT05F001 DD *
=C81075.DAT
/*
//
ENDINPUT
```

Appendix A.2

FILER

```
DSNAME='IJL.FILER.PLI'
000010  FILER: PROC(STARTING_DATE) OPTIONS (MAIN) REORDER;
000020  /*************************************************************************
000030  *                                                                       *
000040  *   THIS PROGRAM READS THE HOURLY DATA FROM THE UT-ACES DATA             *
000050  *   ACQUISITION SYSTEM AND REWRITES THE DATA INTO A REGIONAL             *
000060  *   ORGANIZATION FOR USE BY THE VARIOUS DATA REDUCTIONS PROGRAMS.        *
000070  *   THE FOLLOWING IS A LIST OF THE OUTPUT RECORD NUMBERS AND THEIR       *
000080  *   MEANING.                                                            *
000090  *                                                                       *
000100  *                                                                       *
000110  *   RECORD       MEANING                                                *
000120  *                                                                       *
000130  *                                                                       *
000140  *      1 ----- ACES ICE STORAGE BIN HEAT OUTPUT (WH)                    *
000150  *                                                                       *
000160  *      2 ----- ACES ICE STORAGE BIN HEAT INPUT (WH)                     *
000170  *                                                                       *
000180  *      3 ----- ACES SOLAR PANEL HEAT OUTPUT (WH)                        *
000190  *                                                                       *
000200  *      4 ----- ACES SOLAR PANEL HEAT INPUT (WH)                         *
000210  *                                                                       *
000220  *      5 ----- ACES COOLING PUMP WATT HOUR METER (WH)                   *
000230  *                                                                       *
000240  *      6 ----- ACES HEATING PUMP WATT HOUR METER (WH)                   *
000250  *                                                                       *
000260  *      7 ----- ACES DOMESTIC HOT WATER WATT HOUR METER (WH)             *
000270  *                                                                       *
000280  *      8 ----- ACES COMPRESSOR WATT HOUR METER (WH)                     *
000290  *                                                                       *
000300  *      9 ----- ACES FAN WATT HOUR METER (WH)                            *
000310  *                                                                       *
000320  *     10 ----- ACES HOUSE TOTAL WATT HOUR METER (WH)                    *
000330  *                                                                       *
000340  *     11 ----- ACES FAN COIL OUTPUT (WH)                                *
000350  *                                                                       *
000360  *     12 ----- ACES FAN COIL INPUT (WH)                                 *
000370  *                                                                       *
000380  *     13 ----- ACES DOMESTIC HOT WATER OUTPUT (WH)                      *
000390  *                                                                       *
000400  *     14 ----- ACES ECONOMY CYCLE SECONDS DH                            *
000410  *                                                                       *
000420  *     15 ----- SPARE                                                    *
000430  *                                                                       *
000440  *     16 ----- ACES ECONOMY CYCLE SECONDS H27                           *
000450  *                                                                       *
000460  *     17 ----- SPARE                                                    *
000470  *                                                                       *
000480  *     18 ----- SPARE                                                    *
000490  *                                                                       *
000500  *     19 ----- SOLAR AIR HANDLER WATT HOUR METER (WH)                   *
000510  *                                                                       *
000520  *   RECORD       MEANING                                                *
000530  *                                                                       *
000540  *     20 ----- HEAT PUMP INDOOR (WH)
```

```
000550 *                                                                *
000560 *     21 ----- HEAT PUMP OUTDOOR (WH)                            *
000570 *                                                                *
000580 *     22 ----- OFF PEAK HEATER (WH)
000590 *                                                                *
000600 *     23 ----- HEAT PUMP AUXILIARY HEATER (WH)                   *
000610 *                                                                *
000620 *     24 ----- SOLAR COLLECTOR - SPACE (WH)
000630 *                                                                *
000640 *     25 ----- HEAT PUMP - SPACE (WH)
000650 *                                                                *
000660 *     26 ----- SOLAR COLLECTOR PEBBLE BED (WH)                   *
000670 *                                                                *
000680 *     27 ----- PEBBLE BED - SPACE
000690 *                                                                *
000700 *     28 ----- TOTAL POWER (WH)                                  *
000710 *                                                                *
000720 *     29 ----- SPARE                                             *
000730 *                                                                *
000740 *     30 ----- SPARE
000750 *                                                                *
000760 *     31 ----- SOLAR COLLECTER HEAT FLOW (WH)                    *
000770 *                                                                *
000780 *     32 ----- SOLAR WATER TO REFRIGERANT COIL HEAT FLOW (WH)
000790 *                                                                *
000800 *     33 ----- SOLAR WATER TO AIR COIL HEAT FLOW (WH)
000810 *                                                                *
000820 *     34 ----- SOLAR HOT WATER HEATER OUTPUT (WH)                 *
000830 *                                                                *
000840 *     35 ----- SOLAR HOUSE TOTAL WATT HOUR METER (WH)
000850 *                                                                *
000860 *     36 ----- SOLAR BLOWER WATT HOUR METER (WH)
000870 *                                                                *
000880 *     37 ----- SOLAR HOT WATER HEATER WATT HOUR METER (WH)       *
000890 *                                                                *
000900 *     38 ----- SOLAR COLLECTOR PUMP WATT HOUR METER (WH)         *
000910 *                                                                *
000920 *     39 ----- SOLAR DUCT HEATER WATT HOUR METER (WH)            *
000930 *                                                                *
000940 *     40 ----- SOLAR HEATING PUMP WATT HOUR METER (WH)           *
000950 *                                                                *
000960 *     41 ----- SOLAR HEAT PUMP WATT HOUR METER (WH)
000970 *                                                                *
000980 *     42 ----- SOLAR REFRIGERANT TO AIR COIL HEAT FLOW (WH)
000990 *                                                                *
001000 *     43 ----- SPARE                                             *
001010 *                                                                *
001020 *     44 ----- SPARE                                             *
001030 *                                                                *
001040 *     45 ----- SPARE                                             *
001050 *                                                                *
001060 *   RECORD       MEANING                                         *
001070 *                                                                *
001080 *     46 ----- SOLAR RADIATION 45 DEG (WH/M/M)                   *
001090 *                                                                *
001100 *     47 ----- OUTSIDE WIND SPEED (KPH)                          *
001110 *                                                                *
```

```
001120 *     48 ----- CONTROL HOUSE WATT HOUR METER (WH)
001130 *                                                                *
001140 *     49 ----- CONTROL HOUSE HOT WATER HEATER WATT HOUR METER (WH)
001150 *                                                                *
001160 *     50 ----- CONTROL HOUSE INDOOR UNIT WATT HOUR METER (WH)
001170 *                                                                *
001180 *     51 ----- CONTROL HOUSE OUTDOOR UNIT WATT HOUR METER (WH)
001190 *                                                                *
001200 *     52 ----- CONTROL HOUSE FAN COIL INPUT (WH)
001210 *                                                                *
001220 *     53 ----- CONTROL HOUSE FAN COIL OUTPUT (WH)
001230 *                                                                *
001240 *     54 ----- SPARE                                             *
001250 *                                                                *
001260 *     55 ----- SPARE                                             *
001270 *                                                                *
001280 *     56 ----- SPARE                                             *
001290 *                                                                *
001300 *     57 ----- SPARE                                             *
001310 *                                                                *
001320 *     58 ----- SPARE                                             *
001330 *                                                                *
001340 *     59 ----- SPARE                                             *
001350 *                                                                *
001360 *     60 ----- SPARE                                             *
001370 *                                                                *
001380 *     61 ----- ACES ICE STORAGE BIN LEVEL (KG)                   *
001390 *                                                                *
001400 *     62 ----- ACES ICE STORAGE BIN TEMPERATURE LOW (C)          *
001410 *                                                                *
001420 *     63 ----- ACES BIN TEMPERATURE HIGH (C)
001430 *                                                                *
001440 *     64 ----- ACES ICE STORAGE BIN COIL INLET TEMPERATURE (C)
001450 *                                                                *
001460 *     65 ----- ACES ICE STORAGE BIN COIL OUTLET TEMPERATURE (C)
001470 *                                                                *
001480 *     66 ----- ACES FAN COIL TEMPERATURE INLET (C)
001490 *                                                                *
001500 *     67 ----- ACES FAN COIL TEMPERATURE OUTLET (C)
001510 *                                                                *
001520 *     68 ----- ACES SOLAR PANEL TEMPERATURE INLET (C)
001530 *                                                                *
001540 *     69 ----- ACES SOLAR PANEL TEMPERATURE OUTLET (C)
001550 *                                                                *
001560 *     70 ----- ACES DOMESTIC HOT WATER TEMPERATURE INLET (C)
001570 *                                                                *
001580 *     71 ----- ACES DOMESTIC HOT WATER TEMPERATURE OUTLET (C)
001590 *                                                                *
001600 *   RECORD       MEANING                                         *
001610 *                                                                *
001620 *     72 ----- ACES INSIDE DRY BULB TEMPERATURE (C)              *
001630 *                                                                *
001640 *     73 ----- ACES INSIDE HUMIDITY RATIO (KGW/KGA)              *
001650 *                                                                *
001660 *     74 ----- ACES DRY BULB TEMPERATURE AT SOLAR PANEL (C)
001670 *                                                                *
001680 *     75 ----- SPARE                                             *
001690 *                                                                *
```

```
001700 *     76 ----- SPARE                                                *
001710 *                                                                   *
001720 *     77 ----- SPARE                                                *
001730 *                                                                   *
001740 *     78 ----- SPARE                                                *
001750 *                                                                   *
001760 *     79 ----- SPARE                                                *
001770 *                                                                   *
001780 *     80 ----- PEBBLE BED #1 TEMPERATURE (F)
001790 *                                                                   *
001800 *     81 ----- PEBBLE BED #2 TEMPERATURE (F)
001810 *                                                                   *
001820 *     82 ----- PEBBLE BED #3 TEMPERATURE (F)
001830 *                                                                   *
001840 *     83 ----- PEBBLE BED #4 TEMPERATURE (F)
001850 *                                                                   *
001860 *     84 ----- PEBBLE BED #5 TEMPERATURE (F)
001870 *                                                                   *
001880 *     85 ----- PEBBLE BED #6 TEMPERATURE (F)
001890 *                                                                   *
001900 *     86 ----- PEBBLE BED #7 TEMPERATURE (F)
001910 *                                                                   *
001920 *     87 ----- GARAGE DRY BULB TEMPERATURE (F)
001930 *                                                                   *
001940 *     88 ----- GARAGE WET BULB TEMPERATURE (F)
001950 *                                                                   *
001960 *     89 ----- SPARE                                                *
001970 *                                                                   *
001980 *     90 ----- SPARE                                                *
001990 *                                                                   *
002000 *     91 ----- SPARE                                                *
002010 *                                                                   *
002020 *     92 ----- SPARE                                                *
002030 *                                                                   *
002040 *     93 ----- SPARE                                                *
002050 *                                                                   *
002060 *     94 ----- SPARE                                                *
002070 *                                                                   *
002080 *     95 ----- SPARE                                                *
002090 *                                                                   *
002100 *     96 ----- RAINFALL (MM)                                        *
002110 *                                                                   *
002120 *     97 ----- OUTSIDE WIND DIRECTION (DEG)                         *
002130 *                                                                   *
002140 *   RECORD       MEANING                                            *
002150 *                                                                   *
002160 *     98 ----- OUTSIDE BAROMETRIC PRESSURE (MM-OF-HG)               *
002170 *                                                                   *
002180 *     99 ----- OUTSIDE HUMIDITY RATIO (KGW/KGA)                     *
002190 *                                                                   *
002200 *    100 ----- OUTSIDE DRY BULB TEMPERATURE (C)                     *
002210 *                                                                   *
002220 *    101 ----- SPARE                                                *
002230 *                                                                   *
002240 *    102 ----- SOLAR PANEL INLET TEMPERATURE (C)
002250 *                                                                   *
002260 *    103 ----- SOLAR PANEL OUTLET TEMPERATURE (C)
```

```
002270 *                                                                *
002280 *    104 ----- SOLAR HOT WATER STORAGE TANK TEMPERATURE TOP (C)   *
002290 *                                                                *
002300 *    105 ----- SOLAR PANEL UNPUMPED TEMPERATURE (C)            *
002310 *                                                                *
002320 *    106 ----- SOLAR PANEL PUMPED TEMPERATURE (C)             *
002330 *                                                                *
002340 *    107 ----- SOLAR HOT WATER STORAGE TANK TEMPERATURE BOTTOM (C)  *
002350 *                                                                *
002360 *    108 ----- SOLAR COLLECTOR TANK #2 HIGH (C)                  *
002370 *                                                                *
002380 *    109 ----- SOLAR COLLECTOR TANK #2 LOW (C)                   *
002390 *                                                                *
002400 *    110 ----- SOLAR COLLECTOR TANK #1 HIGH (C)                  *
002410 *                                                                *
002420 *    111 ----- SOLAR COLLECTOR TANK #1 LOW (C)                   *
002430 *                                                                *
002440 *    112 ----- SOLAR COLLECTOR UNIT TEMPERATURE INLET (C)
002450 *                                                                *
002460 *    113 ----- SOLAR HEATING UNIT TEMPERATURE OUTLET (C)          *
002470 *                                                                *
002480 *    114 ----- SOLAR INSIDE DRY BULB TEMPERATURE (C)             *
002490 *                                                                *
002500 *    115 ----- SOLAR INSIDE HUMIDITY RATIO (KGW/KGA)             *
002510 *                                                                *
002520 *    116 ----- SPARE                                             *
002530 *                                                                *
002540 *    117 ----- SPARE                                             *
002550 *                                                                *
002560 *    118 ----- SPARE                                             *
002570 *                                                                *
002580 *    119 ----- SPARE                                             *
002590 *                                                                *
002600 *    120 ----- SPARE                                             *
002610 *                                                                *
002620 *    121 ----- SPARE                                             *
002630 *                                                                *
002640 *    122 ----- SPARE                                             *
002650 *                                                                *
002660 *    123 ----- SPARE                                             *
002670 *                                                                *
002680 *   RECORD       MEANING                                         *
002690 *                                                                *
002700 *    124 ----- SPARE                                             *
002710 *                                                                *
002720 *    125 ----- SPARE                                             *
002730 *                                                                *
002740 *    126 ----- SPARE                                             *
002750 *                                                                *
002760 *    127 ----- SPARE                                             *
002770 *                                                                *
002780 *    128 ----- SPARE                                             *
002790 *                                                                *
002800 *    129 ----- SPARE                                             *
002810 *                                                                *
002820 *    130 ----- SPARE                                            *
002830 *                                                                *
```

```
002840 *    131 ----- SPARE                                                *
002850 *                                                                   *
002860 *    132 ----- SPARE                                                *
002870 *                                                                   *
002880 *    133 ----- SPARE                                                *
002890 *                                                                   *
002900 *    134 ----- SPARE                                                *
002910 *                                                                   *
002920 *    135 ----- SPARE                                                *
002930 *                                                                   *
002940 *    136 ----- CONTROL HOUSE FAN COIL INLET TEMPERATURE (C)
002950 *                                                                   *
002960 *    137 ----- CONTROL HOUSE FAN COIL OUTLET TEMPERATURE (C)
002970 *                                                                   *
002980 *    138 ----- CONTROL HOUSE DRY BULB TEMPERATURE (C)               *
002990 *                                                                   *
003000 *    139 ----- CONTROL HOUSE HUMIDITY RATIO (KGW/KGA)              *
003010 *                                                                   *
003020 *    140 ----- CONTROL HOUSE CRAWL SPACE TEMPERATURE (C)           *
003030 *                                                                   *
003040 *    141 ----- ACES TOTAL SYSTEM KWH (WH)                          *
003050 *                                                                   *
003060 *    142 ----- SOLAR RADIATION 30 DEG (WH/M/M)                     *
003070 *                                                                   *
003080 *    143 ----- SOLAR RADIATION  HORIZONTAL (WH/M/M)                *
003090 *                                                                   *
003100 *    144 ----- OUTSIDE AIR SPECIFIC VOLUME (M*M*M/KG)
003110 *                                                                   *
003120 *    145 ----- ECONOMY CYCLE COOLING LOAD (WH)
003130 *                                                                   *
003140 *    146 ----- SPARE                                                *
003150 *                                                                   *
003160 *    147 ----- SPARE                                                *
003170 *                                                                   *
003180 *    148 ----- SPARE                                                *
003190 *                                                                   *
003200 *    149 ----- SPARE                                                *
003210 *                                                                   *
003220 *   RECORD       MEANING                                            *
003230 *                                                                   *
003240 *    150 ----- SPARE                                                *
003250 *                                                                   *
003260 *    151 ----- SPARE                                                *
003270 *                                                                   *
003280 *    152 ----- SPARE                                                *
003290 *                                                                   *
003300 *    153 ----- SPARE                                                *
003310 *                                                                   *
003320 *    154 ----- SPARE                                                *
003330 *                                                                   *
003340 *    155 ----- SPARE                                                *
003350 *                                                                   *
003360 *    156 ----- SPARE                                                *
003370 *                                                                   *
003380 *    157 ----- SPARE                                                *
003390 *                                                                   *
003400 *    158 ----- INTEGRATION TIME FOR SCAN (SEC)
```

```
003410 *                                                                *
003420 *    159 ----- MONTH DAY                          *
003430 *                                                                *
003440 *    160 ----- HOUR MINUTE
003450 */
003460    DCL (STARTING_TIME,DAY,HOUR,READER,TIME,REAL_TIME,ERROR,I,KGO,
003470         THE_DAY,EXPERIMENT,IMS,SORT) FIXED BINARY(31),
003480         STARTING_DATE CHAR(100) VARYING,
003490        PLIXOPT CHAR(100) STATIC EXTERNAL INIT('ISA(14K') VARYING,
003500         II FIXED BINARY(31),
003510         NUM(12) CHAR(1) INIT('-',' ','0','1','2','3','4','5','6','7',
003520            '8','9'),
003530        EDABC(5) FLOAT DEC(6),
003540        (CLOUD_COVER,DIFFUSE,DIRECT) FLOAT DEC(6),
003550        (ARG,CDEC,CH,CLAT,COSW,COSZ,COST,H,IDN,L,L_30,L_45,LMT,
003560         LONG,M_45,M,N,N_30,N_45,PRO,RATIO1,SDEC,SH,SLAT,SN,TDEC,
003570         TLAT) FLOAT DEC(6),
003580        TABLE(6,13) FLOAT DEC(6) INIT (21.,52.,80.,111.,141.,172.,
003590                    202.,233.,264.,294.,325.,355.,386.,-11.3,-13.5,
003600                    -7.3,.1,3.5,-1.8,-6.4,-1.3,7.5,15.1,13.9,1.5,
003610                    11.3,-20.,-10.8,0.,11.6,20.,23.45,20.6,12.3,0.,
003620                    -10.5,-19.8,-23.45,-20.,390.,385.,376.,360.,350.,
003630                    345.,344.,351.,365.,378.,387.,391.,390.,.142,.144,
003640                    .156,.18,.196,.205,.207,.201,.177,.16,.149,.142,
003650                    .142,.058,.06,.071,.097,.121,.134,.136,.122,.092,
003660                    .073,.063,.057,.058),
003670        (ABS,SQRT) BUILTIN,
003680        KEY FIXED DEC(5),
003690        (IY,IM,ID,IYS) FIXED DEC(6),
003700        MONTH(0:12) FIXED DEC(6) INIT (0,31,60,91,121,152,182,213,244,
003710                                       274,305,335,366),
003720        (EXP,MOD,COS,SIN,LOG10,INDEX) BUILTIN,
003730        SORTOUT FILE EXT,
003740        PRINT FILE PRINT EXT,
003750        ACESOUT FILE OUTPUT KEYED ENV(REGIONAL(1)) EXT,
003760        (ICARD1,ICARD2) FIXED BINARY(31),
003770                CARD_960 CHAR(960),
003780        SUBSTR BUILTIN,
003790        OUTPUT_DATA(160,24) FLOAT DEC(6),
003800        INPUT_DATA(160) FLOAT DEC (6),
003810        1 OUTPUT,
003820          2 DATA(24) FLOAT DEC(6);
003830      SORT=INDEX(STARTING_DATE,'NOSORT');
003840      II=INDEX(STARTING_DATE,'EXP'); EXPERIMENT=0;
003850      IF II>0 THEN DO; II=II-2; EXPERIMENT=1;
003860         STARTING_DATE=SUBSTR(STARTING_DATE,1,II);
003870      END;
003880 /* DETERMINE THE STARTING DATE AS PASSED IN FROM THE PARM CARD */
003890      DAY,HOUR,KEY=0; STARTING_DATE=SUBSTR(STARTING_DATE,2,5);
003900      ON CONVERSION BEGIN; REVERT CONVERSION;
003910        PUT FILE(PRINT) EDIT ('THE STRING "',STARTING_DATE,
003920            '" IS NOT A VALID DATE.') (SKIP(2),A(12),A(5),A(22));
003930        STOP; END;
003940      STARTING_TIME=STARTING_DATE; REVERT CONVERSION;
003950      PUT FILE(PRINT) EDIT ('THE STARTING DATE FOR THIS FILE IS -- ',
003960              STARTING_DATE,'.') (SKIP(2),A(38),A(5),A(1));
003970 /* PUT THE STARTING YEAR, DAY, AND MONTH ON RECORD 0 OF NEW FILE */
```

```
003980     IYS,IY=STARTING_TIME/1000; ID=STARTING_TIME-IY*1000;
003990         REVERT CONVERSION;
004000     IYS=IYS*100000;
004010     DO IM=1 TO 12;
004020       IF ID <= MONTH(IM) THEN GO TO LOOPOUT;
004030     END;
004040     LOOPOUT: ID=ID-MONTH(IM-1); DATA(1)=IY; DATA(2)=IM; DATA(3)=ID;
004050      DATA(4)=EXPERIMENT; IMS=IM;
004060     WRITE FILE (ACESOUT) FROM (OUTPUT) KEYFROM (KEY);
004070  /* SORT THE INPUT DATA BASED ON THE DATE-TIME GROUP IN
004080     ASCENDING ORDER */
004090     CALL SORTER(SORT); OPEN FILE(SORTOUT);
004100     STARTING_TIME=STARTING_TIME*100;
004110     READER=0; KGO=1; CALL SOLAR_DAY; CALL HELI;
004120  /* PROCESS THE SORTED DATA */
004130     DO DAY=1 TO 7;
004140       THE_DAY=THE_DAY+1; CALL SOLAR; CALL DHELI;
004150       DO HOUR=1 TO 24;
004160         IF READER = 1 THEN GO TO SKIP_READ;
004170         READ:
004180         ON ENDFILE (SORTOUT) BEGIN; CALL END_DATA; GO TO SKIP_READ; END;
004190  /* READ THE INPUT DATA */
004200        ON CONVERSION GO TO BAD_DATA;
004210           GET FILE(SORTOUT) EDIT (CARD_960) (A(960));
004220        DO ICARD1= 1 TO  160;
004230       INPUT_DATA(ICARD1)=SUBSTR(CARD_960,ICARD1*6-5,6);
004240          GO TO END_OF_LOOP;
004250  BAD_DATA:
004260        INPUT_DATA(ICARD1)=99999.;
004270    END_OF_LOOP:
004280    END; REVERT CONVERSION;
004290         IF INPUT_DATA(159)+INPUT_DATA(160)>=99999. THEN GO TO READ;
004300         CALL TIMER; IF TIME < 0 THEN GO TO READ;
004310         SKIP_READ:
004320         REAL_TIME=(DAY-1)*100+HOUR;
004330  /* CHECK FOR MISSING DATA AND INSERT 99999. IF FOUND */
004340         IF TIME < REAL_TIME THEN GO TO READ;
004350         IF TIME > REAL_TIME THEN DO; READER=1;
004360                                      OUTPUT_DATA(*,HOUR)=99999.;
004370                                      OUTPUT_DATA(160,HOUR)=REAL_TIME;
004380                                      ERROR=1; CALL ERROR_ROUTINE;
004390                                      GO TO END_HOUR_LOOP; END;
004400         READER=0;
004410         DO I=1 TO 160;
004420       /*   CHECK WETBULB READING HIGHER THAN DRYBULB READING  */
004430    IF INPUT_DATA(99) >= INPUT_DATA(100) THEN INPUT_DATA(99)=
004440                INPUT_DATA(100)-1.0;
004450  /* EXTERNAL TEMPERATURES                */
004460    IF INPUT_DATA(99) < -40. | INPUT_DATA(99) > 50.
004470        THEN INPUT_DATA(99) = 99999.;
004480    IF INPUT_DATA(100) < -30. | INPUT_DATA(100) > 50.
004490        THEN INPUT_DATA(100) = 99999.;
004500  /* INTERNAL TEMPERATURES                */
004510    IF INPUT_DATA(72) < 5. | INPUT_DATA(115) < 5. | INPUT_DATA(139)
004520               < 5. THEN   INPUT_DATA(72) = 99999. | INPUT_DATA(115) =
004530                99999. | INPUT_DATA(139) = 99999.;
004540      IF INPUT_DATA(72) > 50. | INPUT_DATA(115) > 50. |
```

```
004550          INPUT_DATA(139) > 50.  THEN  INPUT_DATA(72) = 99999. ;
004560          INPUT_DATA(115) = 99999. ¦ INPUT_DATA(139) = 99999.;
004570   /*  BAROMETRIC PRESSURE      */
004580       IF INPUT_DATA(98) < 686. ¦ INPUT_DATA(98) = 826.
004590        THEN    INPUT_DATA(98) = 99999.;
004600         OUTPUT_DATA(I,HOUR)=INPUT_DATA(I);
004610         END;
004620  /* CALCULATE TOTAL ACES CONSUMPTION */
004630     OUTPUT_DATA(141,HOUR)=OUTPUT_DATA( 5,HOUR)+OUTPUT_DATA( 6,HOUR)
004640                 +OUTPUT_DATA( 7,HOUR)+OUTPUT_DATA( 8,HOUR)
004650                 +OUTPUT_DATA( 9,HOUR);
004660     IF OUTPUT_DATA(141,HOUR) > 99999. THEN OUTPUT_DATA(141,HOUR)=99999.;
004670    IF INPUT_DATA(158)=99999.  THEN OUTPUT_DATA(159,HOUR)=99999.;
004680   ELSE OUTPUT_DATA(159,HOUR)=INPUT_DATA(158);
004690         OUTPUT_DATA(160,HOUR)=REAL_TIME;
004700         IF OUTPUT_DATA(46,HOUR)=99999. THEN DO;
004710            OUTPUT_DATA(142,HOUR),OUTPUT_DATA(143,HOUR)=99999.;
004720            GO TO WET1; END;
004730  /* CALCULATE SOLAR CLOUD COVER */
004740         CALL HELIOS_45;
004750         IF OUTPUT_DATA(46,HOUR)=0. THEN DO; OUTPUT_DATA(142,HOUR)=0.;
004760            OUTPUT_DATA(143,HOUR)=0.; GO TO WET1; END;
004770           IF OUTPUT_DATA(46,HOUR)>= DIRECT+DIFFUSE  THEN
004780             CLOUD_COVER=0;   ELSE
004790         CLOUD_COVER=SQRT((1.-OUTPUT_DATA(46,HOUR)/
004800              (DIRECT+DIFFUSE))/.65);
004810  /* CALCULATE SOLAR RADIATION ON SURFACE 30 DEG FROM HORIZONTAL */
004820         CALL HELIOS_30;
004830     OUTPUT_DATA(142,HOUR)=(DIRECT+DIFFUSE)*(1.-.65*
004840                            CLOUD_COVER**2);
004850  /* CALCULATE SOLAR RADIATION ON HORIZONTAL SURFACE */
004860         CALL HELIOS_HORIZONTAL;
004870     OUTPUT_DATA(143,HOUR)=(DIRECT+DIFFUSE)*(1.-.65*
004880                            CLOUD_COVER**2);
004890          WET1:
004900         IF OUTPUT_DATA(72,HOUR)=99999. THEN DO;
004910          OUTPUT_DATA(73,HOUR)=99999.; GO TO WET2; END;
004920         IF OUTPUT_DATA(73,HOUR)=99999. THEN GO TO WET2;
004930       IF OUTPUT_DATA(98,HOUR)=99999. THEN DO;
004940          OUTPUT_DATA(73,HOUR)=99999.; GO TO WET2; END;
004950  /* CONVERT FROM WET BULB TO HUMIDITY RATIO */
004960         CALL WET_BULB;
004970          WET2:
004980         IF OUTPUT_DATA(138,HOUR)=99999. THEN DO;
004990          OUTPUT_DATA(139,HOUR)=99999.; GO TO WET22; END;
005000         IF OUTPUT_DATA(139,HOUR)=99999. THEN GO TO WET22;
005010       IF OUTPUT_DATA(98,HOUR)=99999. THEN DO;
005020          OUTPUT_DATA(139,HOUR)=99999.; GO TO WET22; END;
005030  /* CONVERT FROM WET BULB TO HUMIDITY RATIO */
005040         CALL WET_BULB3;
005050          WET22:
005060          IF OUTPUT_DATA(114,HOUR)=99999. THEN DO;
005070             OUTPUT_DATA(115,HOUR)=99999.; GO TO WET3; END;
005080          IF OUTPUT_DATA(114,HOUR)=99999. THEN GO TO WET3;
005090          IF OUTPUT_DATA(98,HOUR)=99999. THEN DO;
005100             OUTPUT_DATA(114,HOUR)=99999.; GO TO WET3; END;
005110  /* CONVERT FROM WET BULB TO HUMIDITY RATIO */
```

```
005120          CALL WET_BULB1;
005130          WET3:
005140          IF OUTPUT_DATA(100,HOUR)=99999. THEN DO;
005150             OUTPUT_DATA(99,HOUR)=99999.; GO TO WET4; END;
005160          IF OUTPUT_DATA(99,HOUR)=99999. THEN GO TO WET4;
005170          IF OUTPUT_DATA(98,HOUR)=99999. THEN DO;
005180             OUTPUT_DATA(99,HOUR)=99999.; GO TO WET4; END;
005190  /* CONVERT FROM DEW POINT TO HUMIDITY RATIO */
005200          CALL WET_BULB2;
005210          WET4:
005220          IF OUTPUT_DATA(72,HOUR)=99999. | OUTPUT_DATA(73,HOUR)=99999.|
005230             OUTPUT_DATA(100,HOUR)=99999. | OUTPUT_DATA(99,HOUR)=99999. |
005240             OUTPUT_DATA(14,HOUR)=99999. | OUTPUT_DATA(144,HOUR)=99999. |
005250             OUTPUT_DATA(16,HOUR)=99999. | OUTPUT_DATA(14,HOUR)=99999.
005260             THEN OUTPUT_DATA(145,HOUR)=99999.;
005270             ELSE OUTPUT_DATA(145,HOUR)=((0.24*OUTPUT_DATA(72,HOUR)
005280                 +OUTPUT_DATA(73,HOUR)*(0.444*OUTPUT_DATA(72,HOUR)
005290                 +1061.))-(0.24*OUTPUT_DATA(100,HOUR)
005300                 +OUTPUT_DATA(99,HOUR)*(0.444*OUTPUT_DATA(100,HOUR)
005310                 +1061.)))*(OUTPUT_DATA(14,HOUR)+OUTPUT_DATA(16,HOUR))
005320                 *1050./OUTPUT_DATA(144,HOUR);
005330       END_HOUR_LOOP: END;
005340  /* CHECK FOR UNREASONABLE DATA */
005350     CALL REASONABLE_DATA(OUTPUT_DATA);
005360  /* OUTPUT DATA FOR PERMANENT STORAGE */
005370       DO I=1 TO 160;
005380         DATA(*)=OUTPUT_DATA(I,*); KEY=KEY+1;
005390         WRITE FILE (ACESOUT) FROM (OUTPUT) KEYFROM (KEY);
005400       END;
005410  /* TERMINATE THE PROGRAM */
005420     END; ERROR=0; CALL ERROR_ROUTINE;
005430  %PAGE;
005440    WET_BULB: PROC REORDER;
005450  /* THIS ROUTINE PERFORMS CONVERSION FROM WET BULB AND DEW POINT
005460     TO HUMIDITY RATIO */
005470     DCL H_RATIO FLOAT DEC(6),(D1,D2,D3,D4) FLOAT DEC(6);
005480         D1=OUTPUT_DATA(98,HOUR)/25.4  ;
005490         D2=(OUTPUT_DATA(72,HOUR))*9./5.+32.0;
005500         D3=(OUTPUT_DATA(73,HOUR))*9./5.+32.0;
005510       CALL PSY(D2,D3,D1,H_RATIO);
005520   /*  CALL PSY (OUTPUT_DATA(72,HOUR),OUTPUT_DATA(73,HOUR),
005530               OUTPUT_DATA(98,HOUR),H_RATIO)*/
005540        OUTPUT_DATA(73,HOUR)=H_RATIO;
005550     RETURN;
005560     WET_BULB1: ENTRY;
005570         D1=OUTPUT_DATA(98,HOUR)/25.4  ;
005580         D2=(OUTPUT_DATA(114,HOUR))*9./5.+32.0;
005590         D3=(OUTPUT_DATA(115,HOUR))*9./5.+32.0;
005600    /*  CALL PSY (OUTPUT_DATA(114,HOUR),OUTPUT_DATA(115,HOUR),
005610               OUTPUT_DATA(98,HOUR),H_RATIO)*/
005620       CALL PSY(D2,D3,D1,H_RATIO);
005630     OUTPUT_DATA(115,HOUR)=H_RATIO;
005640     RETURN;
005650     WET_BULB2: ENTRY;
005660      D1=OUTPUT_DATA(98,HOUR)/25.4;
005670      D2=(OUTPUT_DATA(100,HOUR))*9./5.+32.0;
005680      D3=(OUTPUT_DATA(99,HOUR))*9./5.+32.0;
005690      CALL DEW_POINT (D2,D3,D1,H_RATIO,D4);
```

```
005700  /*  CALL DEW_POINT (OUTPUT_DATA(100,HOUR),OUTPUT_DATA(99,HOUR),
005710            OUTPUT_DATA(98,HOUR),H_RATIO,OUTPUT_DATA(144,HOUR))*/
005720     OUTPUT_DATA(99,HOUR)=H_RATIO;
005730        OUTPUT_DATA(144,HOUR)=D4*0.062426;
005740     RETURN;
005750     WET_BULB3: ENTRY;
005760         D1=OUTPUT_DATA(98,HOUR)/25.4  ;
005770         D2=(OUTPUT_DATA(114,HOUR))*9./5.+32.0;
005780         D3=(OUTPUT_DATA(115,HOUR))*9./5.+32.0;
005790  /*  CALL PSY (OUTPUT_DATA(114,HOUR),OUTPUT_DATA(115,HOUR),
005800       OUTPUT_DATA(98,HOUR),H_RATIO)*/
005810       CALL PSY(D2,D3,D1,H_RATIO);
005820     OUTPUT_DATA(139,HOUR)=H_RATIO;
005830     RETURN;
005840    END WET_BULB;
005850  %PAGE;
005860    DEW_POINT: PROC (DB,DP,PB,W,V) REORDER;
005870  /* THIS ROUTINE CONVERTS FROM DEW POINT TO HUMIDITY RATIO */
005880     DCL (PV,PVS,RH,W,DB,DP,PB) FLOAT DEC(6);
005890     PV=PVSF(DP); PVS=PVSF(DB); RH=PV/PVS; W=.622*PV/(PB-PV);
005900     V=.754*(DB+459.7)*(1.+7000.*W/4360.)/PB;
005910    END DEW_POINT;
005920  %PAGE;
005930    PSY: PROC (DB,WB,PB,W) REORDER;
005940  /* THIS ROUTINE CONVERTS FROM WET BULB TO HUMIDITY RATIO */
005950     DCL (DB,WB,PB,W,V,PVP,WSTAR,PV,CDB,CWB,HL,CH,EX) FLOAT DEC(6);
005960     PVP=PVSF(WB);
005970     IF DB-WB<= 0. THEN DO; PV=PVP; GO TO A; END;
005980     WSTAR=0.622*PVP/(PB-PVP);
005990     IF WB-32.<= 0. THEN DO; PV=PVP-5.704E-4*PB*(DB-WB)/1.8;
006000                             GO TO A; END;
006010     CDB=(DB-32.)/1.8; CWB=(WB-32.)/1.8;
006020     HL=597.31+0.4409*CDB-CWB; CH=0.2402+0.4409*WSTAR;
006030     EX=(WSTAR-CH*(CDB-CWB)/HL)/0.622; PV=PB*EX/(1.+EX);
006040     A:
006050     W=0.622*PV/(PB-PV);
006060     RETURN;
006070    END PSY;
006080  %PAGE;
006090    PVSF: PROC (X) REORDER;
006100  /* THIS ROUTINE DETERMINES PARTIAL PRESSURES */
006110     DCL (X,T,Z,Z1,SUM) FLOAT DEC(6),
006120         I FIXED BINARY (31),
006130         A(6) FLOAT DEC(6) INIT (-7.9028,5.02808,-1.3816E-7,
006140                                 11.344,8.1328E-3,-3.49149),
006150         B(4) FLOAT DEC(6) INIT (-9.09718,-3.56654,0.876793,
006160                                 0.0060273),
006170         P(4) FLOAT DEC(6);
006180     T=(X+459.7)/1.8;
006190     IF T<273.16 THEN DO; Z=273.16/T; P(1)=B(1)*(Z-1.);
006200                          P(2)=B(2)*LOG10(Z); P(3)=B(3)*(1.-1./Z);
006210                          P(4)=LOG10(B(4)); END;
006220                 ELSE DO; Z=373.16/T; P(1)=A(1)*(Z-1.);
006230                          P(2)=A(2)*LOG10(Z); Z1=A(4)*(1.-1./Z);
006240                          P(3)=A(3)*(10.**Z1-1.); Z1=A(6)*(Z-1.);
006250                          P(4)=A(5)*(10.**Z1-1.); END;
006260     SUM=0.; DO I=1 TO 4; SUM=SUM+P(I); END;
```

```
006270     RETURN (29.921*10.**SUM);
006280    END PVSF;
006290  %PAGE;
006300    TIMER: PROC REORDER;
006310  /* THIS ROUTINE CONVERTS FROM REAL TIME TO RELATIVE TIME */
006320     DCL (CHECK,TIME1) FIXED BINARY(31),
006330         MONTH(12) FIXED BINARY(31) INIT (0,3100,6000,9100,12100,
006340                                          15200,18200,21300,24400,
006350                                          27400,30500,33500);
006360     CHECK=INPUT_DATA(159)/100; TIME=(INPUT_DATA(159)-CHECK*100)*100;
006370     TIME=TIME+MONTH(CHECK)+IYS+INPUT_DATA(160)/100;
006380     TIME1=STARTING_TIME/100000; TIME1=STARTING_TIME-TIME1*100000;
006390     IF TIME1<35500 THEN GO TO A;
006400     IF CHECK<IMS THEN TIME=TIME+36500;
006410    A: TIME=TIME-STARTING_TIME+1;
006420     IF DAY+HOUR = 2 THEN GO TO START;
006430     RETURN;
006440     START:
006450     IF TIME > 624 THEN DO; ERROR=2; CALL ERROR_ROUTINE; END;
006460     RETURN;
006470    END TIMER;
006480  %PAGE;
006490    SOLAR: PROC REORDER;
006500  /* THIS ROUTINE DETERMINES THE SUN'S POSITION */
006510     DCL (DAY,K,KEY,KEYP1) FIXED BINARY(31);
006520     DAY=THE_DAY; IF DAY <= TABLE(1,1) THEN DO; DAY=DAY+365.; KEY=12;
006530                                                KEYP1=13; END;
006540                  ELSE DO; DO K=KGO TO 13;
006550                       IF TABLE(1,K) > DAY THEN DO; KEYP1=K;
006560                                KGO,KEY=K-1; GO TO A; END;
006570                            END; A: END;
006580     DO K=2 TO 6;
006590       EDABC(K-1)=TABLE(K,KEY)+(TABLE(K,KEYP1)-TABLE(K,KEY))*
006600                  (DAY-TABLE(1,KEY))/(TABLE(1,KEYP1)-TABLE(1,KEY));
006610     END; RETURN;
006620     SOLAR_DAY: ENTRY;
006630     DO K=1 TO 13; TABLE(3,K)=TABLE(3,K)/57.2958; END;
006640     THE_DAY=(STARTING_TIME-IYS)/100-1;
006650     RETURN;
006660    END SOLAR;
006670  %PAGE;
006680    HELIOS: PROC REORDER;
006690  /* THIS ROUTINE DETERMINES THE SUN'S HOUR ANGLE AND THE SOLAR
006700     RADIATION INCIDENT ON A SURFACE */
006710     DCL (DT,COSS,ALPHA) FLOAT DEC(6);
006720     HELIOS_HORIZONTAL: ENTRY;
006730     COST=COSZ;
006740     IF COST < 0. THEN COST=0.;
006750     DIRECT=IDN*COST*3.1525;
006760     RETURN;
006770     HELIOS_30: ENTRY;
006780     COST=L_30*COSZ+N_30*COSS+M*COSW;
006790     IF COST < 0. THEN COST=0.;
006800     DIRECT=IDN*COST*3.1525;
006810     RETURN;
006820     HELIOS_45: ENTRY;
006830     DIRECT,DIFFUSE=0.; LMT=HOUR-1-LONG/15.; DT=ABS(LMT-SN);
006840     H=15.*DT/57.2958;
```

```
006850     SH=SIN(H); CH=COS(H); COSZ=SLAT*SDEC+CLAT*CDEC*CH; COSW=CDEC*SH;
006860     ARG=1.-COSW*COSW-COSZ*COSZ; COSS=0.;
006870     IF ARG > 0. THEN COSS=SQRT(ARG);
006880     IF CH <= RATIO1 THEN COSS=-COSS;
006890     IDN=0.;
006900     IF CH > -PRO THEN DO; IF COSZ > 0.01 THEN IDN=EDABC(3)/
006910                           EXP(EDABC(4)/COSZ); END;
006920     COST=L_45*COSZ+N_45*COSS+M_45*COSW;
006930     IF COST < 0. THEN COST=0.;
006940     DIRECT=IDN*COST*3.1525; DIFFUSE=IDN*EDABC(5)*3.1525;
006950     RETURN;
006960     HELI: ENTRY;
006970     LONG=84.; L=1.; N=0.; M=0.; L_45=.707;
006980     N_45=.7018; LONG=MOD(LONG+7.49,15.)-7.49; SLAT=.61566; CLAT=.788;
006990     TLAT=.7813; ALPHA=0.; L_30=.866; N_30=0.5; M_45=.0862;
007000     RETURN;
007010     DHELI: ENTRY;
007020     SN=12.-EDABC(1)/60.; SDEC=SIN(EDABC(2)); CDEC=COS(EDABC(2));
007030     TDEC=SDEC/CDEC; RATIO1=TDEC/TLAT; PRO=TDEC/TLAT;
007040     RETURN;
007050   END HELIOS;
007060 %PAGE;
007070   END_DATA: PROC REORDER;
007080 /* THIS ROUTINE HANDLES THE END OF DATA CONDITION */
007090     IF DAY+HOUR = 2 THEN DO; ERROR=3; CALL ERROR_ROUTINE; END;
007100     TIME=1000;
007110     RETURN;
007120   END END_DATA;
007130 %PAGE;
007140   ERROR_ROUTINE: PROC REORDER;
007150 /* THIS ROUTINE HANDLES VARIOUS ERROR CONDITIONS AND TERMINATES
007160     THE JOB WHEN APPLICABLE */
007170     DCL ERROR_ONE STATIC INIT (0) FIXED BINARY(31),
007180       PLIRETC BUILTIN,
007190       RETC FIXED BINARY(31),
007200         (ERROR_TWO,ERROR_THREE) FIXED BINARY(31);
007210     ERROR_TWO,ERROR_THREE=0;
007220     IF ERROR=0 THEN GO TO ZERO;
007230     IF ERROR = 1 THEN GO TO ONE;
007240     IF ERROR = 2 THEN GO TO TWO;
007250     IF ERROR = 3 THEN GO TO THREE;
007260     ONE:
007270     ERROR_ONE=ERROR_ONE+1; RETC=4; CALL PLIRETC(RETC);
007280     PUT FILE (PRINT) EDIT ('ASH001 - THERE IS NO DATA FOR DAY = ',
007290                            DAY,' HOUR = ',HOUR,' .',
007300                            'STANDARD FIXUP TAKEN ---- ',
007310                            'EXECUTION CONTINUING.')
007320                           (SKIP(2),A,F(10),A,F(10),A,SKIP,X(9),A,A);
007330     RETURN;
007340     TWO:
007350     ERROR_TWO=1; RETC=8; CALL PLIRETC(RETC);
007360     PUT FILE (PRINT) EDIT ('ASH002 --- PROGRAM IS TERMINATING DUE TO ',
007370                            'ERROR COUNT ON ERROR 2.','THERE IS NO ',
007380                            'DATA FOR THE WEEK CHOSEN ON THE INPUT ',
007390                            'DATASET.')
007400                           (PAGE,A,A,SKIP,X(11),A,A,A);
007410     DATA(*)=99999.;
```

```
007420     DO KEY=1 TO 1120;
007430        WRITE FILE(ACESOUT) FROM (OUTPUT) KEYFROM (KEY);
007440     END;
007450     GO TO ZERO;
007460     THREE:
007470     ERROR_THREE=1; RETC=12; CALL PLIRETC(RETC);
007480     PUT FILE (PRINT) EDIT ('ASH003 --- PROGRAM IS TERMINATING DUE TO ',
007490                            'ERROR COUNT FOR ERROR 3.','           END ',
007500                            'OF DATASET WAS ENCOUNTERED BEFORE ',
007510                            'REACHING STARTING DATE.')
007520                            (PAGE,A,A,SKIP,A,A,A);
007530     DATA(*)=99999.;
007540     DO KEY=1 TO 1120;
007550        WRITE FILE(ACESOUT) FROM (OUTPUT) KEYFROM (KEY);
007560     END;
007570     ZERO:
007580     IF DAY+HOUR = 2 THEN GO TO NONE;
007590     PUT FILE (PRINT) EDIT ('THERE WERE ',ERROR_ONE,' EMPTY RECORDS ',
007600                            'PROCESSED.  DATASET WAS CREATED.')
007610                           (PAGE,A,F(10),A,A);
007620     GO TO SUMMARY;
007630     NONE:
007640     PUT FILE (PRINT) EDIT ('DATASET WAS CREATED BUT IS EMPTY')
007650              (PAGE,A);
007660     SUMMARY:
007670     IF ERROR_ONE+ERROR_TWO+ERROR_THREE = 0 THEN GO TO FINISH;
007680     PUT FILE (PRINT) EDIT ('S U M M A R Y   O F   E R R O R S',
007690                            'ERROR NUMBER          COUNT','')
007700                           (PAGE,X(10),A,SKIP(2),X(13),A,SKIP,A);
007710     IF ERROR_ONE > 0 THEN PUT FILE (PRINT) EDIT ('001',ERROR_ONE)
007720                                            (SKIP,X(17),A,X(14),F(5));
007730     IF ERROR_TWO = 1 THEN PUT FILE (PRINT) EDIT ('002',ERROR_TWO)
007740                                            (SKIP,X(17),A,X(14),F(5));
007750     IF ERROR_THREE = 1 THEN PUT FILE (PRINT) EDIT ('003',ERROR_THREE)
007760                                              (SKIP,X(17),A,X(14),F(5));
007770     FINISH:
007780     CLOSE FILE (ACESOUT); GO TO END_OF_FILER;
007790   END ERROR_ROUTINE;
007800  %PAGE;
007810   REASONABLE_DATA: PROC(DATA) REORDER;
007820  /* THIS ROUTINE DETERMINES IF THE DATA IS REASONABLE AND
007830     RESPONDS ACCORDINGLY */
007840   DCL DATA(160,24) FLOAT DEC(6),
007850       (HIGH(160),LOW(160),REPLACE(160)) FLOAT DEC(6),
007860       (I,J) FLOAT BINARY(31);
007870      I=1;  IF I=1 THEN GO TO WORK;   RETURN;
007880     WORK:
007890      LOW(*)=-40.; REPLACE(*)=99999.; HIGH(*)=300.;
007910        DO I=1 TO 10;
007920           LOW(I)=0.;
007930        END;
007950           DO I = 11,12;
007960              LOW(I)=50.0; END;
007961         LOW(13),LOW(14),LOW(15),LOW(16)=0.0;
007970              HIGH(1)=22000.; HIGH(2)=10000.;
007980              HIGH(3)=20000.; HIGH(4)=HIGH(1);
007990              HIGH(5)=180.; HIGH(6)=170.;
008000              HIGH(7)=100.; HIGH(8)=3600.;
```

```
008010              HIGH(9)=550.; HIGH(10)=6500.;
008020              HIGH(11)=HIGH(2); HIGH(12)=15000.;
008030              HIGH(13)=8000.; HIGH(14)=HIGH(8);
008040              HIGH(16)=HIGH(14);
008050  /*    GARAGE   EXPERIMENT  DATA */
008060            DO I=19 TO 28;
008070                  LOW(I)=0.;
008080            END;
008090              HIGH(19),HIGH(20)=2000.;
008100              HIGH(21),HIGH(22),HIGH(23)=20000.;
008110              HIGH(24),HIGH(26),HIGH(27)=30000.;
008120              HIGH(25),HIGH(28)=50000.;
008130    /*              SOLAR HOUSE      */
008140              DO I= 31 TO 42;
008150                LOW(I)= 0.;
008160              END;
008170               DO I=48 TO 51;
008180                  LOW(I)=0.0;
008190                END ;
008200                HIGH(31)=150000.;   HIGH(32)=8000.;
008210                HIGH(33)=10000.; HIGH(34)=HIGH(32);
008220                HIGH(35)=14000.; HIGH(36)=550.;
008230                HIGH(37)=4000.; HIGH(38)=1500.;
008240                HIGH(39)=8500.; HIGH(40)=500.;
008250                HIGH(41)=5000.; HIGH(42)=HIGH(33) ;
008260  /*     CONTROL HOUSE */
008270               HIGH(46)=1300.; HIGH(47)=70.;
008280               HIGH(48)=27000.; HIGH(49)=5000.;
008290               HIGH(50)=13000.; HIGH(51)=HIGH(49);
008300               HIGH(62)=15.; LOW(62)=-2.; HIGH(63)=HIGH(62);
008310               LOW(63)=LOW(62); HIGH(64),HIGH(65)=5.0;
008320               LOW(64),LOW(65)=-7.;
008330               HIGH(66),HIGH(67)=45.0; LOW(66),LOW(67)=30.;
008340               HIGH(68),HIGH(69)=60.; LOW(68),LOW(69)=0.;
008350        /*                                  */
008360               HIGH(70),HIGH(71)=55.; LOW(70),LOW(71)=8.;
008380               HIGH(73)=20.; LOW(73)=0.;
008390               HIGH(74)=150.; LOW(74)=-30.;
008400  /*    CONTINUED GARAGE  EXPERIMENT DATA */
008410            DO I=80 TO 88;
008420                LOW(I)=-15.0;
008430                HIGH(I)=200.0;
008440            END;
008450         /*                                  */
008490               HIGH(102),HIGH(103)=100.; LOW(102),LOW(103)=0.;
008500               HIGH(104),HIGH(107)=55.; LOW(104),LOW(107)=8.;
008510               HIGH(105),HIGH(106)=150.; LOW(105),LOW(106)=-30.;
008520         /*                                  */
008530               DO I =108 TO 111;
008540                  LOW(I)=2.; HIGH(I)=100.;
008550                END;
008560               HIGH(112),HIGH(113)=100.; LOW(112),LOW(113)=0.;
008570                HIGH(114)=30.;  LOW(114)=5.;
008590         /*                                  */
008600               HIGH(138)=HIGH(72); LOW(138)=LOW(72);
008620               HIGH(140)=45.; LOW(140)=-20.;
008630               HIGH(141)=4000.; LOW(141)=0.;
008640               HIGH(142),HIGH(143)=1300.; LOW(142),LOW(143)=0.;
```

```
008650              HIGH(145)=HIGH(33); LOW(145)=0.;
008660                 DO I =1 TO 160;
008670                     DO J = 1 TO 24;
008700                        IF DATA(I,J)>HIGH(I) THEN DATA(I,J)=REPLACE(I);
008710                        IF DATA(I,J)<LOW(I) THEN DATA(I,J)=REPLACE(I);
008720                      END;
008730                   END;
008740    RETURN;
008750   END REASONABLE_DATA;
008760  %PAGE;
008770   SORTER: PROC(SORT) REORDER;
008780  /* THIS ROUTINE SORTS THE INPUT DATA INTO ASCENDING ORDER BASED
008790     ON THE DATE-TIME GROUP */
008800   DCL ACESIN EXT FILE,
008810       CARD_1120 CHAR(1120) VARYING,
008820       CARD_960 CHAR(960) VARYING,
008830       (RETURN_CODE,SORT) FIXED BINARY(31),
008840       PLISRTA BUILTIN,
008850       SORTIN EXT FILE;
008860    ON ENDFILE(ACESIN) GO TO END_OF_SORT;
008870    READ:
008880    CARD_960='';
008890    GET FILE(ACESIN) EDIT (CARD_1120) (A(1120));
008900    DO RETURN_CODE=1 TO 13;
008910       CARD_960=CARD_960||SUBSTR(CARD_1120,1,72);
008920       CARD_1120=SUBSTR(CARD_1120,81,LENGTH(CARD_1120)-80);
008930    END;
008940    CARD_960=CARD_960||SUBSTR(CARD_1120,1,24);
008950    IF SORT>0 THEN PUT FILE(SORTOUT) EDIT (CARD_960) (A(960));
008960       ELSE PUT FILE(SORTIN) EDIT (CARD_960)  (A(960));
008970    GO TO READ;
008980    END_OF_SORT: CLOSE FILE(ACESIN);
008990    IF SORT>0 THEN DO; CLOSE FILE(SORTOUT); RETURN; END;
009000    CLOSE FILE(SORTIN);
009010    CALL PLISRTA(' SORT FIELDS=(949,6,CH,A,955,6,CH,A) ',
009020                 ' RECORD TYPE=F,LENGTH=(960) ',51200,RETURN_CODE);
009030    PUT FILE(PRINT) EDIT ('SORT RETURN CODE = ',RETURN_CODE,'')
009040            (SKIP(2),A,F(5),SKIP,A);
009050    RETURN;
009060   END SORTER;
009070  %PAGE;
009080  END_OF_FILER: END FILER;
END OF DATA
READY
```

Appendix A.3

SUMMARY

```
DSNAME='IJL.SUMMARY.PLI'
000010  /******** A N N U A L   C Y C L E   E N E R G Y   S Y S T E M ********/
000020  /*
000030
000040     J C L   R E Q U I R E D   T O   R U N   T H I S   P R O G R A M
000050
000060   //ACESSUM  EXEC PGM=SUMMARY,PARM='ISA(20K,4K,4)/OPTION',REGION=270K
000070   //STEPLIB  DD DSN=PLI.LINKLIB,UNIT=2314,VOL=SER=MVT21Q,DISP=SHR
000080   //         DD DSN=HOLMAN.PRIVATE.LIBRARY,UNIT=2314,VOL=SER=UTACE1,
000090   //            DISP=(OLD,KEEP)
000100   //SYSPRINT DD SYSOUT=A
000110   //SYSPRT1  DD SYSOUT=A
000120   //SYSPRT2  DD SYSOUT=A
000130   //SYSPRT3  DD SYSOUT=A
000140   //PLOTS    DD UNIT=SYSDA,DSN=&&PLOTDATA,SPACE=(CYL,(1,1),RLSE),
000150   //            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
000160   //TOTAL    DD DSN=ACES.DATA.TOTALS,UNIT=2314,
000170   //            VOL=SER=UTACE1,DISP=(OLD,KEEP)
000180   //ACEDATA  DD DSN=ACES.DATA.DXXXXX,UNIT=2314,
000190   //            VOL=SER=UTACE1,DISP=(OLD,KEEP)
000200   //
000210
000220
000230  */
000240  SUMMARY: PROC(OPTION) OPTIONS(MAIN) REORDER;
000250   DCL ACEDATA EXT FILE RECORD DIRECT INPUT KEYED ENV(REGIONAL(1)),
000260       SYSPRINT EXT FILE PRINT,
000270       OPTION CHAR(100) VARYING,
000280       PRIORITY BUILTIN,
000290       TASK_1 TASK,
000300       (EP1,EP2,EP3) EVENT,
000310       (CODE1,CODE2) FIXED BINARY(31),
000320       (MAX,PLIRETC) BUILTIN,
000330       RELATIVE FIXED BINARY(15);
000340    OPEN FILE(SYSPRINT) LINESIZE(132); OPEN FILE(ACEDATA);
000350  /*
000360     SET TASK_1 TO THE MAXIMUM ALLOWABLE PRIORITY AND SET RELATIVE TO
000370     A SMALLER PRIORITY BECAUSE TASK ANALIZE IS A LONG EXECUTING
000380     TASK AND NEEDS HIGHER PRIORITY TO FINISH AT THE SAME TIME AS
000390     TASKS PLOTPLI AND MISSING.
000400  */
000410    PRIORITY(TASK_1)=20; RELATIVE=PRIORITY(TASK_1)-1;
000420    CALL ANALIZE TASK(TASK_1) EVENT(EP1);
000430    CALL MISSING(CODE1) PRIORITY(RELATIVE) EVENT(EP2);
000440    CALL PLOTPLI(OPTION,CODE2) PRIORITY(RELATIVE) EVENT(EP3);
000450  /*
000460     WAIT FOR THE COMPLETION OF ALL THE TASKS BEFORE TERMINATING THE
000470     PROGRAM.
000480  */
000490    WAIT (EP1,EP2,EP3);
000500    CODE1=MAX(CODE1,CODE2);
000510    CALL PLIRETC(CODE1);
000520  %PAGE;
000530  /*
```

```
000540     MISSING - PRINTS THE HOURS FOR WHICH THERE IS NO DATA AVAILABLE
000550               IN THE ACES DATA FILE.
000560  */
000570  MISSING: PROC(CODE1) OPTIONS(TASK) REORDER;
000580   DCL SYSPRT1 EXT FILE PRINT,
000590       ACEDATA EXT FILE DIRECT INPUT RECORD KEYED ENV(REGIONAL(1)),
000600       CODE1 FIXED BINARY(31),
000610       (COUNT,DAY,HOUR,TABLE(168),PAGE_NO,LINE_LENGTH,I,
000620        RECORD_TABLE(159),RECORD,NUMB,ERROR,ERROR_TABLE(159,168),
000630        J,DATE(3)) FIXED BINARY(15),
000640       1 INPUT_DATA,
000650         2 DATA(24) FLOAT DEC(6),
000660       RECORD_NO FIXED DEC(5);
000670    OPEN FILE(SYSPRT1) LINESIZE(132); CODE1=0;
000680    RECORD_NO=0; READ FILE(ACEDATA) INTO(INPUT_DATA) KEY(RECORD_NO);
000690    DATE(1)=DATA(2); DATE(2)=DATA(3); DATE(3)=DATA(1);
000700    COUNT=0;
000710  /*
000720     SCAN LOOP CHECKS THE INTEGRATION TIME IN RECORD NUMBER 159.
000730     IF THIS NUMBER IS 99999. THEN THERE IS NO DATA FOR THE HOUR.
000740  */
000750    SCAN: DO DAY=1 TO 7;
000760       RECORD_NO=(DAY-1)*160+159;
000770       READ FILE(ACEDATA) INTO(INPUT_DATA) KEY(RECORD_NO);
000780       DO HOUR=1 TO 24;
000790          IF DATA(HOUR)=99999. THEN DO;
000800             COUNT=COUNT+1; TABLE(COUNT)=(DAY-1)*24+HOUR;
000810    END SCAN; PAGE_NO=1;
000820    PUT FILE(SYSPRT1) EDIT ('*****  S U M M A R Y   O F   F I L E   ',
000830             'F O R   ',DATE(1),'/',DATE(2),'/',DATE(3),'  *****',
000840             'PAGE',PAGE_NO,'')
000850            (PAGE,X(35),A,A,F(2),A,F(2),A,F(2),A,X(26),A,F(5),SKIP,A);
000860    PUT FILE(SYSPRT1) EDIT ('*** LISTING OF HOURS FOR WHICH THERE IS ',
000870             'NO DATA ***','') (SKIP(3),X(40),A,A,SKIP,A);
000880    IF COUNT=168 THEN CODE1=4;
000890    IF COUNT=0 THEN
000900       PUT FILE(SYSPRT1) EDIT ('---- THERE ARE NO MISSING HOURS ----')
000910                (SKIP,A);
000920       ELSE DO; LINE_LENGTH=41;
000930       PUT FILE(SYSPRT1) EDIT ('---- THE FOLLOWING HOURS ARE MISSING ',
000940                '----') (SKIP(3),A,A);
000950       DO I=1 TO COUNT;
000960          LINE_LENGTH=LINE_LENGTH+5;
000970          IF LINE_LENGTH>132 THEN DO; LINE_LENGTH=46;
000980             PUT FILE(SYSPRT1) EDIT ('') (SKIP,X(41),A); END;
000990          IF I=COUNT THEN
001000             PUT FILE(SYSPRT1) EDIT (TABLE(I)) (F(4));
001010                   ELSE
001020             PUT FILE(SYSPRT1) EDIT (TABLE(I),',') (F(4),A);
001030    END; END; ERROR=0; RECORD_TABLE(*)=0;
001040  /*
001050     OTHER LOOPS CHECKS FOR MISSING DATA IN THE INDIVIDUAL RECORDS
001060     NOT ALREADY INCLUDED IN THE SCAN CHECK.
001070  */
001080    OTHER: DO DAY=1 TO 7;
001090       RECORD_NO=(DAY-1)*160;
001100       DO RECORD=1 TO 159;
001110          RECORD_NO=RECORD_NO+1;
```

```
001120          READ FILE(ACEDATA) INTO(INPUT_DATA) KEY(RECORD_NO);
001130          DO HOUR=1 TO 24;
001140             IF DATA(HOUR)=99999. THEN GO TO LABEL;
001150             GO TO LABEL_B;
001160             LABEL: NUMB=(DAY-1)*24+HOUR; IF COUNT=0 THEN GO TO LABEL_A;
001170             DO I=1 TO COUNT;
001180                IF NUMB=TABLE(I) THEN GO TO LABEL_B;
001190             END;
001200          LABEL_A: RECORD_TABLE(RECORD)=RECORD_TABLE(RECORD)+1;
001210          ERROR=1; ERROR_TABLE(RECORD,RECORD_TABLE(RECORD))=NUMB;
001220    LABEL_B: END OTHER; RECORD=0;
001230    ON ENDPAGE(SYSPRT1) BEGIN; PAGE_NO=PAGE_NO+1;
001240       PUT FILE(SYSPRT1) EDIT ('*****  S U M M A R Y   O F   F I L E   ',
001250                'F O R   ',DATE(1),'/',DATE(2),'/',DATE(3),'  *****',
001260                'PAGE',PAGE_NO,'')
001270               (PAGE,X(35),A,A,F(2),A,F(2),A,F(2),A,X(26),A,F(5),SKIP,A);
001280       PUT FILE(SYSPRT1) EDIT ('*** LISTING OF HOURS MISSING FOR ',
001290                'INDIVIDUAL RECORDS ***',
001300                '(DOES NOT INCLUDE THOSE LISTED PREVIOUSLY)','')
001310               (SKIP(2),X(38),A,A,SKIP,X(45),A,SKIP,A);
001320    END; SIGNAL ENDPAGE(SYSPRT1);
001330    IF ERROR=0 THEN DO;
001340       PUT FILE(SYSPRT1) EDIT ('---- THERE ARE NO ADDITIONAL HOURS ',
001350                'MISSING ----') (SKIP(3),A,A);
001360       GO TO LABEL_D;
001370    END;
001380  /*
001390     PUT1 LOOP OUTPUTS THE MISSING RECORDS.
001400  */
001410    PUT1: DO RECORD=1 TO 159;
001420       LINE_LENGTH=37;
001430       PUT FILE(SYSPRT1) EDIT ('---- HOURS MISSING FOR RECORD',RECORD,
001440                ' ----') (SKIP,A,F(3),A);
001450       IF RECORD_TABLE(RECORD)=0 THEN
001460          PUT FILE(SYSPRT1) EDIT ('NO HOURS ARE MISSING FOR THIS RECORD')
001470                  (A);
001480          ELSE DO;
001490          DO J=1 TO RECORD_TABLE(RECORD);
001500             LINE_LENGTH=LINE_LENGTH+5;
001510             IF LINE_LENGTH>132 THEN DO; LINE_LENGTH=42;
001520                PUT FILE(SYSPRT1) EDIT ('') (SKIP,X(37),A); END;
001530             PUT FILE(SYSPRT1) EDIT
001540                     (ERROR_TABLE(RECORD,J),',')
001550                     (F(4),A);
001560    END PUT1;
001570    LABEL_D: CLOSE FILE(SYSPRT1); RETURN;
001580  END MISSING;
001590  %PAGE;
001600  /*
001610     PLOTPLI - PRODUCES THE PLOT DATA FOR THE PLOTTING PROGRAM
001620               AND PASSES THE RESULTS IN FILE PLOTS.
001630  */
001640  PLOTPLI: PROC(OPTION,CODE1) OPTIONS(TASK) REORDER;
001650   DCL PLOTS EXT FILE STREAM OUTPUT,
001660       OPTION CHAR(100) VARYING,
001670       ACEDATA EXT FILE INPUT DIRECT KEYED RECORD ENV(REGIONAL(1)),
001680       SYSPRT2 EXT FILE PRINT,
001690       (I,J,CODE1) FIXED BINARY(31),
```

```
001700       (IM,REC,ID,IY) FIXED BINARY(15),
001710       CARD_23 CHAR(23),
001720       CARD_49 CHAR(49),
001730       PLOT CHAR(7) VARYING,
001740       (Y_ARRAY(168),YT_ARRAY(168)) FLOAT DEC(6),
001750       (YTT_ARRAY(168),YTTT_ARRAY(168),YTTTT_ARRAY(168),
001760        YTTTTTT_ARRAY(168)) FLOAT DEC(6),
001770       (CHAR,INDEX) BUILTIN,
001780       IKEY FIXED DEC(5),
001790       1 INPUT,
001800         2 DATA(24) FLOAT DEC(6),
001810     CARD_80 CHAR(80);
001820     OPEN FILE(PLOTS); OPEN FILE(SYSPRT2) LINESIZE(120); CARD_49=(49) ' ';
001830     IKEY=0; READ FILE(ACEDATA) INTO (INPUT) KEY(IKEY);
001840     IY=DATA(1); IM=DATA(2); ID=DATA(3);
001850     I=INDEX(OPTION,'BOTH'); IF I > 0 THEN PLOT='BOTH';
001860     I=INDEX(OPTION,'METRIC'); IF I > 0 THEN PLOT='METRIC';
001870     I=INDEX(OPTION,'ENGLISH'); IF I > 0 THEN PLOT='ENGLISH';
001880     I=INDEX(OPTION,'NOPLOT'); IF I > 0 THEN PLOT='NOPLOT';
001890     I=INDEX(OPTION,'BOTH')+INDEX(OPTION,'METRIC')
001900       +INDEX(OPTION,'ENGLISH')+INDEX(OPTION,'NOPLOT');
001910     IF I = 0 THEN DO;
001920        PLOT='NOPLOT';
001930        PUT FILE(SYSPRT2) EDIT ('INVALID OPTION IN PARM FIELD FOR ',
001940                 'PLOT=','NOPLOTS ASSUMED','')
001950                (PAGE,A,A,SKIP,A,SKIP,A); END;
001960     IF PLOT='NOPLOT' THEN GO TO END_OF_PLOTPLI;
001970     IF PLOT='BOTH' THEN DO; CARD_23='BOTH'||'(W)EEK (B)EGINNING ';
001980        PUT FILE(PLOTS) EDIT (CARD_23,IM,'/',ID,'/',IY,CARD_49)
001990            (A(23),F(2),A(1),F(2),A(1),F(2),A(49)); END;
002000     IF PLOT='METRIC' THEN DO; CARD_23='MET '||'(W)EEK (B)EGINNING ';
002010        PUT FILE(PLOTS) EDIT (CARD_23,IM,'/',ID,'/',IY,CARD_49)
002020            (A(23),F(2),A(1),F(2),A(1),F(2),A(49)); END;
002030     IF PLOT='ENGLISH' THEN DO; CARD_23='ENG '||'(W)EEK (B)EGINNING ';
002040        PUT FILE(PLOTS) EDIT (CARD_23,IM,'/',ID,'/',IY,CARD_49)
002050            (A(23),F(2),A(1),F(2),A(1),F(2),A(49)); END;
002060     REC=100; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002070     REC=99; CALL READ_CARDS(YT_ARRAY,REC);
002080     CALL WET_BULB(Y_ARRAY,YT_ARRAY); CALL WRITE_CARDS(Y_ARRAY);
002090     REC=72; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002100     REC=73; CALL READ_CARDS(YT_ARRAY,REC);
002110     CALL WET_BULB(Y_ARRAY,YT_ARRAY); CALL WRITE_CARDS(Y_ARRAY);
002120     REC=142; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002130      CALL LOAD1(Y_ARRAY); CALL WRITE_CARDS(Y_ARRAY);
002140      CALL LOAD2(YT_ARRAY); CALL WRITE_CARDS(YT_ARRAY);
002150      CALL LOAD3(YTT_ARRAY); CALL WRITE_CARDS(YTT_ARRAY);
002160      CALL LOAD4(YTTT_ARRAY); CALL WRITE_CARDS(YTTT_ARRAY);
002170      CALL LOAD5(YTTTTTT_ARRAY); CALL WRITE_CARDS(YTTTTTT_ARRAY);
002180     REC=47; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002190     REC=143; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002200     REC=4; CALL READ_CARDS(Y_ARRAY,REC);
002210     REC=3; CALL READ_CARDS(YT_ARRAY,REC);
002220     DO I=1 TO 168;
002230        Y_ARRAY(I)=Y_ARRAY(I)+YT_ARRAY(I);
002240        IF Y_ARRAY(I) > 99999. THEN Y_ARRAY(I)=99999.;
002250     END; YT_ARRAY(*)=0.; CALL WRITE_CARDS(Y_ARRAY);
002260     DO I=5 TO 7;
002270        REC=I; CALL READ_CARDS(Y_ARRAY,REC);
```

```
002280      DO J=1 TO 168;
002290         YT_ARRAY(J)=YT_ARRAY(J)+Y_ARRAY(J);
002300         IF YT_ARRAY(J) > 99999. THEN YT_ARRAY(J)=99999.;
002310      END;
002320   END; CALL WRITE_CARDS(YT_ARRAY);
002330   DO I=8,9,141,10;
002340      REC=I; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002350   END; YT_ARRAY(*)=0.;
002360   REC=8; CALL READ_CARDS(Y_ARRAY,REC);
002370   DO I=1 TO 168;
002380      IF Y_ARRAY(I) < 100. THEN DO; Y_ARRAY(I)=0.; GO TO A1; END;
002390      IF YTTTTTT_ARRAY(I)=99999. THEN DO; Y_ARRAY(I)=0.; GO TO A1; END;
002400      IF Y_ARRAY(I) = 99999. THEN DO; Y_ARRAY(I)=0.; GO TO A1; END;
002410      (NOOFL,NOZDIV,NOUFL): Y_ARRAY(I)=YTTTTTT_ARRAY(I)/Y_ARRAY(I);
002420      A1:
002430      IF Y_ARRAY(I) > 99999. THEN Y_ARRAY(I)=99999.;
002440   END; CALL WRITE_CARDS(Y_ARRAY);
002450   REC=141; CALL READ_CARDS(Y_ARRAY,REC);
002460   DO I=1 TO 168;
002470      IF Y_ARRAY(I) < 100. THEN DO; Y_ARRAY(I)=0.; GO TO A2; END;
002480      IF YTTTTTT_ARRAY(I)=99999. THEN DO; Y_ARRAY(I)=0.; GO TO A2; END;
002490      IF Y_ARRAY(I) = 99999. THEN DO; Y_ARRAY(I)=0.; GO TO A2; END;
002500      (NOOFL,NOZDIV,NOUFL): Y_ARRAY(I)=YTTTTTT_ARRAY(I)/Y_ARRAY(I);
002510      A2:
002520      IF Y_ARRAY(I) > 99999. THEN Y_ARRAY(I)=99999.;
002530   END; CALL WRITE_CARDS(Y_ARRAY); CALL WRITE_CARDS(YTTTTTT_ARRAY);
002540   REC=100; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002550   REC=99; CALL READ_CARDS(YT_ARRAY,REC);
002560   CALL WET_BULB(Y_ARRAY,YT_ARRAY); CALL WRITE_CARDS(Y_ARRAY);
002570   REC=138; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002580   REC=139; CALL READ_CARDS(YT_ARRAY,REC);
002590   CALL WET_BULB(Y_ARRAY,YT_ARRAY); CALL WRITE_CARDS(Y_ARRAY);
002600   /* CHANGED 1-30-79  ADD 48,50,51 FOR TOTAL CON HOUSE EQUIP */
002610   REC=49; CALL READ_CARDS(Y_ARRAY,REC);
002620   /* 2-20-79 MULTIPLY REC=50 BY 2. FOR WEEKS JAN1 JAN8 */
002630   REC=50; CALL READ_CARDS(YT_ARRAY,REC);
002640   REC=51; CALL READ_CARDS(YTT_ARRAY,REC);
002650   DO I=1 TO 168;
002660      YTT_ARRAY(I)=YTT_ARRAY(I)+YT_ARRAY(I)+ Y_ARRAY(I);
002670      IF YTT_ARRAY(I) > 99999. THEN YTT_ARRAY(I)=99999.;
002680   END;
002690   CALL WRITE_CARDS(YTT_ARRAY);
002700   /* ACES  VS CONTROL  HOUSES 2-20 */
002710   REC=141; CALL READ_CARDS(Y_ARRAY,REC); CALL WRITE_CARDS(Y_ARRAY);
002720   CALL WRITE_CARDS(YTT_ARRAY);
002730   CALL WRITE_CARDS(Y_ARRAY); CALL WRITE_CARDS(YTT_ARRAY);
002740   /* ACES VS CONTROL VS SOLAR HOUSES 2-20 */
002750   REC=36; CALL READ_CARDS(Y_ARRAY,REC);
002760   REC=37; CALL READ_CARDS(YT_ARRAY,REC);
002770   REC=38; CALL READ_CARDS(YTT_ARRAY,REC);
002780   REC=39; CALL READ_CARDS(YTTT_ARRAY,REC);
002790   REC=40; CALL READ_CARDS(YTTTT_ARRAY,REC);
002800   DO I=1 TO 168;
002810      Y_ARRAY(I)=Y_ARRAY(I)+YT_ARRAY(I)+YTT_ARRAY(I)+YTTT_ARRAY(I)
002820                 +YTTTT_ARRAY(I);
002830      IF Y_ARRAY(I)>99999. THEN Y_ARRAY(I)=99999.;
002840   END;
002850   CALL WRITE_CARDS(Y_ARRAY);
```

```
002860     REC=51; CALL READ_CARDS(Y_ARRAY,REC);
002870     CALL WRITE_CARDS(Y_ARRAY);
002880     REC=50; CALL READ_CARDS(YT_ARRAY,REC);
002890    /*     DO I= 1 TO 168;
002900         YT_ARRAY(I)=YT_ARRAY(I)*2.;
002910         END;    */
002920     CALL WRITE_CARDS(YT_ARRAY);
002930     REC=49; CALL READ_CARDS(YTT_ARRAY,REC);
002940     CALL WRITE_CARDS(YTT_ARRAY);
002950     DO I=1 TO 168;
002960        Y_ARRAY(I)=YTT_ARRAY(I)+YT_ARRAY(I)+Y_ARRAY(I);
002970         IF Y_ARRAY(I) > 99999. THEN Y_ARRAY(I)=99999.;
002980         END;
002990         CALL WRITE_CARDS(Y_ARRAY);
003000         REC=48; CALL READ_CARDS(YTT_ARRAY,REC);
003010         CALL WRITE_CARDS(YTT_ARRAY);
003020   END_OF_PLOTPLI: CLOSE FILE(SYSPRT2); CLOSE FILE(PLOTS);
003030   CODE1=0; IF PLOT='NOPLOT' THEN CODE1=4;
003040   RETURN;
003050 %PAGE;
003060 /*
003070    U_BAR - PRESENTLY IS NOT BEING PLOTTED.  THE PURPOSE OF THIS
003080            ROUTINE IS TO CALCULATE THE OVERALL HEAT TRANSFER OF
003090            THE SOLAR PANEL
003100 */
003110  U_BAR: PROC(Y_ARRAY) REORDER;
003120  DCL (Y_ARRAY(168),YT_ARRAY(168)) FLOAT DEC(6),
003130      (I,REC) FIXED BINARY(15);
003140   REC=67; CALL READ_CARDS(Y_ARRAY,REC);
003150   REC=100; CALL READ_CARDS(YT_ARRAY,REC);
003160   DO I=1 TO 168;
003170      IF Y_ARRAY(I) = 99999. THEN YT_ARRAY(I)=99999.;
003180      IF YT_ARRAY(I) = 99999. THEN GO TO A;
003190      YT_ARRAY(I)=Y_ARRAY(I)-YT_ARRAY(I); A:
003200   END; REC=44; CALL READ_CARDS(Y_ARRAY,REC);
003210   DO I=1 TO 168;
003220      IF YT_ARRAY(I) = 99999. THEN Y_ARRAY(I)=99999.;
003230      IF Y_ARRAY(I) = 99999. THEN GO TO B;
003240      Y_ARRAY(I)=Y_ARRAY(I)/480./YT_ARRAY(I); B:
003250   END; RETURN;
003260  END U_BAR;
003270 %PAGE;
003280 /*
003290    WET_BULB - CONVERTS FROM HUMIDITY RATIO TO WET BULB
003300 */
003310  WET_BULB: PROC(Y_ARRAY,YT_ARRAY) REORDER;
003320  DCL (Y_ARRAY(168),YT_ARRAY(168),C1) FLOAT DEC(6),
003330      I FIXED BINARY(15),
003340      (EXP,LOG) BUILTIN;
003350   DO I=1 TO 168;
003360      IF YT_ARRAY(I) = 99999. THEN Y_ARRAY(I)=99999.;
003370      IF Y_ARRAY(I) = 99999. THEN GO TO A;
003380      Y_ARRAY(I)=Y_ARRAY(I)*9./5.+32.;
003390      C1=0.24*Y_ARRAY(I)
003400                      +YT_ARRAY(I)*(1061.+0.444*Y_ARRAY(I));
003410      IF C1 <= 0. THEN C1=0.0001;
003420      YT_ARRAY(I)=LOG(C1);
003430      IF EXP(YT_ARRAY(I)) < 11.758
```

```
003440          THEN Y_ARRAY(I)=0.604+3.4841*YT_ARRAY(I)+1.3601*YT_ARRAY(I)**2
003450                          +0.9731*YT_ARRAY(I)**3;
003460          ELSE Y_ARRAY(I)=30.9185-39.682*YT_ARRAY(I)
003470                          +20.5841*YT_ARRAY(I)**2-1.758*YT_ARRAY(I)**3;
003480       Y_ARRAY(I)=(Y_ARRAY(I)-32.)*5./9.;
003490     A: END; RETURN;
003500    END WET_BULB;
003510   %PAGE;
003520   /*
003530      WRITE_CARDS - OUTPUTS THE PLOT INFORMATION TO BOTH THE PLOT FILE
003540                    AND THE LINE PRINTER
003550   */
003560    WRITE_CARDS: PROC(Y_ARRAY) REORDER;
003570    DCL Y_ARRAY(168) FLOAT DEC(6),
003580        FMT(0:2) LABEL,
003590        J FIXED BINARY(31),
003600        I STATIC INIT(0) FIXED BINARY(31);
003610     FMT(0): FORMAT(PAGE,X(50),A,F(5),A,SKIP(2),168 F(10,2));
003620     FMT(1): FMT(2): FORMAT(SKIP(2),X(50),A,F(5),A,SKIP(2),168 F(10,2));
003630     PUT FILE(PLOTS) EDIT (Y_ARRAY) (8 F(10,2));
003640     I=I+1; J=(I-1)/3; J=I-1-J*3;
003650     PUT FILE(SYSPRT2) EDIT ('*** RECORD ',I,' ***',Y_ARRAY)
003660         (R(FMT(J)));
003670    END WRITE_CARDS;
003680   %PAGE;
003690   /*
003700      READ_CARDS - READS THE ACES DATA FILE AND STORES THE VALUES
003710                   IN THE PASSED ARRAY
003720   */
003730    READ_CARDS: PROC(Y_ARRAY,REC) REORDER;
003740    DCL Y_ARRAY(168) FLOAT DEC(6),
003750        IKEY FIXED DEC(5),
003760        (I,J,K,REC) FIXED BINARY(15);
003770     J=-24;
003780     DO I=1 TO 7;
003790        IKEY=(I-1)*160+REC; READ FILE(ACEDATA) INTO (INPUT) KEY (IKEY);
003800        J=J+24;
003810        DO K=1 TO 24;
003820           Y_ARRAY(J+K)=DATA(K);
003830        END;
003840     END; RETURN;
003850    END READ_CARDS;
003860   END PLOTPLI;
003870   %PAGE;
003880   /*
003890      ANALIZE - PRODUCES THE PRINTER OUTPUT FOR THE ACES DATA
003900                INCLUDING PEAK PLOT AND CURVE FIT.
003910   */
003920   ANALIZE: PROC OPTIONS(TASK) REORDER;
003930     DCL ACEDATA EXT FILE RECORD DIRECT INPUT KEYED ENV(REGIONAL(1)),
003940         TOTAL EXT FILE RECORD DIRECT UPDATE KEYED ENV(REGIONAL(1)),
003950         SYSPRT3 EXT FILE PRINT,
003960         RECORD_# FIXED DEC(5),
003970         WEEK_# FIXED DEC(5) INIT(0),
003980         1 INPUT_DATA,
003990           2 DATA(24) FLOAT DEC(6),
004000         1 TOTALS,
004010           2 OLD_DATA(100) FLOAT DEC(6),
```

```
004020        (LINE,LINE_NO,PAGE_#,ID,IM,IY,EXPERIMENT,ZERO) FIXED BINARY(15),
004030        TITLE CHAR(109),
004040        (TRUNC,MAX,LENGTH,SUBSTR,INDEX) BUILTIN,
004050        PRINT FILE VARIABLE,
004060        HEAD_DATE CHAR(38);
004070       PAGE_#,RECORD_#=0; LINE=132; PRINT=SYSPRT3;
004080       OPEN FILE(SYSPRT3) LINESIZE(LINE);
004090       READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
004100       IY=DATA(1); IM=DATA(2); ID=DATA(3); EXPERIMENT=DATA(4);
004110       TITLE='*****  A N N U A L   C Y C L E   E N E R G Y   '||
004120             'S Y S T E M   D E M O N S T R A T I O N   '||
004130             'P R O J E C T  *****';
004140       HEAD_DATE='ANALYZED FOR WEEK BEGINNING - '||CHARS(IM)||
004150                 '/'||CHARS(ID)||'/'||CHARS(IY);
004160                 ZERO=1.;
004170        IF INDEX(OPTION,'ZERO')>0 THEN ZERO=0;
004180       CALL WEEK(EXPERIMENT,ZERO);
004190       CALL ACESSUM;
004200       CALL ENVELOPE;
004210       CALL TIME_OF_DAY;
004220       CALL WEEK(EXPERIMENT,ZERO);
004230  %PAGE;
004240  /*
004250     CHARS - CONVERTS FROM ARITHMETIC TO CHARACTER AND DELETES THE
004260             LEADING BLANKS
004270  */
004280   CHARS: PROC(NUMBER) RETURNS(CHAR(2)) REORDER;
004290    DCL (I,NUMBER) FIXED BINARY(15),
004300        C CHAR(1),
004310        C2 CHAR(2) VARYING,
004320        C1 CHAR(20) VARYING;
004330       C1=NUMBER; C2='';
004340       DO I=1 TO LENGTH(C1);
004350          C=SUBSTR(C1,I,1);
004360          IF C=' ' THEN C=' ';
004370             ELSE C2=C2||C;
004380       END; I=LENGTH(C2);
004390       IF I=1 THEN RETURN('0'||C2);
004400          ELSE RETURN(C2);
004410   END CHARS;
004420  %PAGE;
004430  /*
004440     TIME_OF_DAY - PRODUCES A PRINTER PLOT SHOWING THE TIME OF DAY
004450                   POWER CONSUMPTION OF THE ACES HOUSE VERSUS THE
004460                   CONTROL HOUSE
004470  */
004480   TIME_OF_DAY: PROC REORDER;
004490    DCL (ACES_ARRAY(55,102),CONTROL_ARRAY(55,102)) CHAR(1),
004500        ARRAY(24) FLOAT BINARY,
004510        (I,J,K,L,CODE) FIXED BINARY(15),
004520        PLOT_HEAD CHAR(52);
004530       PLOT_HEAD='WORST DAY POWER USE (0=CONTROL '||
004540                 'HOUSE, 1=ACES HOUSE)';
004550       PRINT=SYSPRT3; CONTROL_ARRAY(*,*)=' ';
004560       CALL TITLING(TITLE,PRINT,LINE,PAGE_#); LINE_NO=0;
004570       CALL HEADING (HEAD_DATE,PRINT,LINE,LINE_NO); LINE_NO=1;
004580       CALL HEADING(PLOT_HEAD,PRINT,LINE,LINE_NO);
004590       ACES_ARRAY(*,*)=' '; ACES_ARRAY(18,1),ACES_ARRAY(30,1)='P';
```

```
004600       ACES_ARRAY(19,1),ACES_ARRAY(25,1),ACES_ARRAY(33,1)='O';
004610       ACES_ARRAY(20,1),ACES_ARRAY(37,1)='W'; ACES_ARRAY(21,1)='E';
004620       ACES_ARRAY(22,1)='R'; ACES_ARRAY(24,1)='C';
004630       ACES_ARRAY(26,1),ACES_ARRAY(34,1)='N'; ACES_ARRAY(27,1)='S';
004640       ACES_ARRAY(28,1)='U'; ACES_ARRAY(29,1)='M';
004650       ACES_ARRAY(31,1)='T'; ACES_ARRAY(32,1)='I';
004660       ACES_ARRAY(36,1)='K'; ACES_ARRAY(38,1)='H';
004670       ACES_ARRAY(55,46)='T'; ACES_ARRAY(55,47)='I';
004680       ACES_ARRAY(55,48)='M'; ACES_ARRAY(55,49)='E';
004690       ACES_ARRAY(55,51)='O'; ACES_ARRAY(55,52)='F';
004700       ACES_ARRAY(55,54)='D'; ACES_ARRAY(55,55)='A';
004710       ACES_ARRAY(55,56)='Y'; ACES_ARRAY(1,4)='0';
004720       ACES_ARRAY(11,4)='8'; ACES_ARRAY(21,4)='6';
004730       ACES_ARRAY(31,4)='4'; ACES_ARRAY(41,4)='2';
004740       ACES_ARRAY(1,3)='1';
004750       ACES_ARRAY(51,4)='0';
004760       DO I=1 TO 10;
004770          ACES_ARRAY(53,I*4+41)='1';
004780       END;
004790       DO I=1 TO 5;
004800          ACES_ARRAY(53,I*4+81)='2';
004810       END;
004820       ACES_ARRAY(53,6),ACES_ARRAY(53,46),ACES_ARRAY(53,86)='0';
004830       ACES_ARRAY(53,10),ACES_ARRAY(53,50),ACES_ARRAY(53,90)='1';
004840       ACES_ARRAY(53,14),ACES_ARRAY(53,54),ACES_ARRAY(53,94)='2';
004850       ACES_ARRAY(53,18),ACES_ARRAY(53,58),ACES_ARRAY(53,98)='3';
004860       ACES_ARRAY(53,22),ACES_ARRAY(53,62),ACES_ARRAY(53,102)='4';
004870       ACES_ARRAY(53,26),ACES_ARRAY(53,66)='5';
004880       ACES_ARRAY(53,30),ACES_ARRAY(53,70)='6';
004890       ACES_ARRAY(53,34),ACES_ARRAY(53,74)='7';
004900       ACES_ARRAY(53,38),ACES_ARRAY(53,78)='8';
004910       ACES_ARRAY(53,42),ACES_ARRAY(53,82)='9';
004920       DO I=2 TO 50;
004930          ACES_ARRAY(I,6),ACES_ARRAY(I,102)='|';
004940       END;
004950       DO I=7 TO 101;
004960          ACES_ARRAY(1,I),ACES_ARRAY(51,I)='-';
004970       END;
004980       DO I=1 TO 51 BY 10;
004990          ACES_ARRAY(I,6),ACES_ARRAY(I,102)='+';
005000       END;
005010       DO I=10 TO 98 BY 4;
005020          ACES_ARRAY(1,I),ACES_ARRAY(51,I)='+';
005030       END; CODE=1;
005040  LL0: CALL PEAK(ARRAY,CODE);
005050  LL1: DO I=1 TO 24;
005060          J=TRUNC(ARRAY(I)*5.+0.5); J=51-J;
005070          IF J<1 THEN J=1;
005080          IF J>50 THEN J=51;
005090             ELSE DO;
005100             DO K=J TO 50;
005110                DO L=I*4+3 TO I*4+5;
005120                   IF CODE=1 THEN ACES_ARRAY(K,L)='1';
005130                             ELSE CONTROL_ARRAY(K,L)='0';
005140       END LL1;
005150       CODE=CODE+1; IF CODE=2 THEN GO TO LL0;
005160       DO I=1 TO 55;
005170          PUT FILE(SYSPRT3) EDIT ((ACES_ARRAY(I,J) DO J=1 TO 102))
```

```
005180                (SKIP,X(12),102 A(1));
005190          PUT FILE(SYSPRT3) EDIT ((CONTROL_ARRAY(I,J) DO J=1 TO 102))
005200                (SKIP(0),X(12),102 A(1));
005210          IF I=11 | I=21 | I=31 | I=41 THEN
005220             PUT FILE(SYSPRT3) EDIT ((95) '-')
005230                   (SKIP(0),X(18),A);
005240       END;
005250       RETURN;
005260  %PAGE;
005270  /*
005280     PEAK - PRODUCES A TABLE OF THE PEAK HOURLY POWER CONSUMPTION
005290            FOR BOTH THE ACES AND CONTROL HOUSES
005300  */
005310   PEAK: PROC(ARRAY,CODE) REORDER;
005320    DCL (CODE,I,DAY) FIXED BINARY(15),
005330        (ARRAY(24),TOTAL,TOTAL1,ARRAY1(24)) FLOAT BINARY;
005340       ARRAY(*)=0.; TOTAL=0.;
005350       DO DAY=1 TO 7;
005360          TOTAL1=0.;
005370          RECORD_#=(DAY-1)*160+5;
005380          IF CODE=2 THEN GO TO LL0;
005390          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005400          CALL CHECK(INPUT_DATA); RECORD_#=RECORD_#+1; ARRAY1=DATA;
005410          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005420          CALL CHECK(INPUT_DATA); RECORD_#=RECORD_#+1;
005430          ARRAY1=ARRAY1+DATA;
005440          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005450          CALL CHECK(INPUT_DATA); RECORD_#=RECORD_#+1;
005460          ARRAY1=ARRAY1+DATA;
005470          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005480          CALL CHECK(INPUT_DATA); RECORD_#=RECORD_#+1;
005490          ARRAY1=ARRAY1+DATA;
005500          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005510          CALL CHECK(INPUT_DATA);
005520          ARRAY1=ARRAY1+DATA; GO TO LL1;
005530     /*  CHANGED 2-5-79   2-20 REC=50*2.  */
005540  LL0:    RECORD_#=RECORD_#+44;
005550          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005560          CALL CHECK(INPUT_DATA); ARRAY1=DATA;
005570  /*      CHANGES MADE 2-5-1979   */
005580          RECORD_#=RECORD_#+1;
005590          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005600          CALL CHECK(INPUT_DATA); ARRAY1=ARRAY1+DATA;
005610          RECORD_#=RECORD_#+1;
005620          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
005630          CALL CHECK(INPUT_DATA); ARRAY1=ARRAY1+DATA;
005640  /*    CHANGES MADE TO ADD RECORDS 48,50,51 FOR CON HOUSE*/
005650  LL1:    DO I=1 TO 24;
005660             TOTAL1=TOTAL1+ARRAY1(I);
005670          END;
005680          IF TOTAL1>TOTAL THEN DO; TOTAL=TOTAL1;
005690             ARRAY=ARRAY1; END;
005700       END;
005710       RETURN;
005720  %PAGE;
005730  /*
005740     CHECK - CONVERTS THE ELECTRICAL CONSUMPTION FROM WATTS TO
005750             KILOWATT-HOURS FOR PEAK.  IF THE VALUE = 99999.
```

```
005760             THEN 0. IS RETURNED
005770  */
005780   CHECK: PROC(INPUT) REORDER;
005790    DCL 1 INPUT,
005800          2 DATA(24)  FLOAT DEC(6),
005810        I FIXED BINARY(15);
005820       DO I=1 TO 24;
005830          IF DATA(I)>=99999. THEN DATA(I)=0.;
005840             ELSE DATA(I)=DATA(I)/1000.;
005850       END;
005860       RETURN;
005870   END CHECK;
005880   END PEAK;
005890   END TIME_OF_DAY;
005900  %PAGE;
005910  /*
005920     ENVELOPE - PRODUCES THE LEAST SQUARES FIT OF THE BUILDING LOADS
005930                VERSUS OUTSIDE AIR TEMPERATURE
005940  */
005950   (NOOVERFLOW,NOUNDERFLOW,NOZERODIVIDE):
005960   ENVELOPE: PROC REORDER;
005970    DCL (COOL_50,COOL_150,COOL_OTHER,HEAT_50,HEAT_150,HEAT_OTHER,
005980         COOL_ALL,HEAT_ALL,DAY,HOUR) FIXED BINARY(15),
005990        HEAD1 CHAR(81),
006000        (SOLAR(24),COOLING(24),HEATING(24),COOL_ARRAY_50(2,168),
006010         COOL_ARRAY_150(2,168),COOL_ARRAY_OTHER(2,168),BREAK_EVEN,
006020         HEAT_ARRAY_50(2,168),HEAT_ARRAY_150(2,168),
006030         COOL_ARRAY_ALL(2,168),HEAT_ARRAY_ALL(2,168),
006040         HEAT_ARRAY_OTHER(2,168),SUM_X,SUM_Y,SUM_X_Y,SUM_X_SQUARED,
006050         Y_INTERCEPT,SLOPE)  FLOAT BINARY;
006060       COOL_50,COOL_150,COOL_OTHER,HEAT_50,HEAT_150,HEAT_OTHER=0;
006070       HEAT_ALL,COOL_ALL=0;
006080  LL0: DO DAY=1 TO 7;
006090          RECORD_#=(DAY-1)*160+143;
006100          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
006110          SOLAR=DATA; RECORD_#=RECORD_#-132;
006120          CALL FILL1 (HEATING,RECORD_#);
006130          RECORD_#=RECORD_#+1;
006140          CALL FILL2 (COOLING,RECORD_#);
006150          RECORD_#=RECORD_#+88;
006160          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
006170          DO HOUR=1 TO 24;
006180             IF DATA(HOUR)=99999. THEN GO TO LL2;
006190             IF COOLING(HOUR)=99999. THEN GO TO LL1;
006200             IF COOLING(HOUR)=0. THEN GO TO LL1;
006210             COOL_ALL=COOL_ALL+1;
006220             COOL_ARRAY_ALL(1,COOL_ALL)=COOLING(HOUR);
006230             COOL_ARRAY_ALL(2,COOL_ALL)=DATA(HOUR);
006240             IF SOLAR(HOUR)<50. THEN DO;
006250                COOL_50=COOL_50+1;
006260                COOL_ARRAY_50(1,COOL_50)=COOLING(HOUR);
006270                COOL_ARRAY_50(2,COOL_50)=DATA(HOUR); GO TO LL2;
006280             END;
006290             IF SOLAR(HOUR)<150. THEN DO;
006300                COOL_150=COOL_150+1;
006310                COOL_ARRAY_150(1,COOL_150)=COOLING(HOUR);
006320                COOL_ARRAY_150(2,COOL_150)=DATA(HOUR); GO TO LL2;
006330             END; COOL_OTHER=COOL_OTHER+1;
```

```
006340              COOL_ARRAY_OTHER(1,COOL_OTHER)=COOLING(HOUR);
006350              COOL_ARRAY_OTHER(2,COOL_OTHER)=DATA(HOUR); GO TO LL2;
006360  LL1:        IF HEATING(HOUR)=99999. THEN GO TO LL2;
006370              IF HEATING(HOUR)=0. THEN GO TO LL2;
006380              HEAT_ALL=HEAT_ALL+1;
006390              HEAT_ARRAY_ALL(1,HEAT_ALL)=HEATING(HOUR);
006400              HEAT_ARRAY_ALL(2,HEAT_ALL)=DATA(HOUR);
006410              IF SOLAR(HOUR)<50. THEN DO;
006420                 HEAT_50=HEAT_50+1;
006430                 HEAT_ARRAY_50(1,HEAT_50)=HEATING(HOUR);
006440                 HEAT_ARRAY_50(2,HEAT_50)=DATA(HOUR); GO TO LL2;
006450              END;
006460              IF SOLAR(HOUR)<150. THEN DO;
006470                 HEAT_150=HEAT_150+1;
006480                 HEAT_ARRAY_150(1,HEAT_150)=HEATING(HOUR);
006490                 HEAT_ARRAY_150(2,HEAT_150)=DATA(HOUR); GO TO LL2;
006500              END; HEAT_OTHER=HEAT_OTHER+1;
006510              HEAT_ARRAY_OTHER(1,HEAT_OTHER)=HEATING(HOUR);
006520              HEAT_ARRAY_OTHER(2,HOUR)=DATA(HOUR);
006530  LL2:     END;
006540        END;
006550        CALL TITLING(TITLE,PRINT,LINE,PAGE_#); LINE_NO=0;
006560        CALL HEADING(HEAD_DATE,PRINT,LINE,LINE_NO); LINE_NO=5;
006570        HEAD1='***** L E A S T   S Q U A R E D   F I T   F O R   '||
006580              'C O O L I N G   L O A D S *****';
006590        CALL HEADING(HEAD1,PRINT,LINE,LINE_NO);
006600        CALL SUM( COOL_ARRAY_50,COOL_50,SUM_X,SUM_Y,SUM_X_Y,
006610                SUM_X_SQUARED);
006620        OLD_DATA(1)=OLD_DATA(1)+COOL_50;
006630        OLD_DATA(2)=OLD_DATA(2)+SUM_X; OLD_DATA(3)=OLD_DATA(3)+SUM_Y;
006640        OLD_DATA(4)=OLD_DATA(4)+SUM_X_Y;
006650        OLD_DATA(5)=OLD_DATA(5)+SUM_X_SQUARED;
006660        PUT FILE(SYSPRT3) EDIT ('SOLAR <  50 (BTU/HR-SQFT):')
006670                (SKIP(3),X(5),A);
006680        IF OLD_DATA(1)<2 THEN PUT FILE(SYSPRT3) EDIT
006690           ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
006700           (X(10),A);
006710           ELSE DO;
006720              CALL CURVE(OLD_DATA(2),OLD_DATA(3),OLD_DATA(4),OLD_DATA(5),
006730                         Y_INTERCEPT,SLOPE,OLD_DATA(1));
006740              PUT FILE(SYSPRT3) EDIT
006750                  ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
006760                   Y_INTERCEPT)
006770                  (X(10),A,E(12,5,6),A,E(12,5,6));
006780              BREAK_EVEN=-Y_INTERCEPT/SLOPE;
006790              PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ",
006800                       BREAK_EVEN,' (C)",'NO. OF POINTS ANALIZED = ",
006810                       OLD_DATA(1))
006820                      (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
006830           END;
006840        CALL SUM( COOL_ARRAY_150,COOL_150,SUM_X,SUM_Y,SUM_X_Y,
006850                SUM_X_SQUARED);
006860        OLD_DATA(6)=OLD_DATA(6)+COOL_150;
006870        OLD_DATA(7)=OLD_DATA(7)+SUM_X; OLD_DATA(8)=OLD_DATA(8)+SUM_Y;
006880        OLD_DATA(9)=OLD_DATA(9)+SUM_X_Y;
006890        OLD_DATA(10)=OLD_DATA(10)+SUM_X_SQUARED;
006900        PUT FILE(SYSPRT3) EDIT ('SOLAR < 150 (BTU/HR-SQFT):')
006910                (SKIP(3),X(5),A);
```

```
006920       IF OLD_DATA(6)<2 THEN PUT FILE(SYSPRT3) EDIT
006930          ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
006940          (X(10),A);
006950          ELSE DO;
006960             CALL CURVE(OLD_DATA(7),OLD_DATA(8),OLD_DATA(9),OLD_DATA(10),
006970                        Y_INTERCEPT,SLOPE,OLD_DATA(6));
006980             PUT FILE(SYSPRT3) EDIT
006990                 ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
007000                  Y_INTERCEPT)
007010                 (X(10),A,E(12,5,6),A,E(12,5,6));
007020             BREAK_EVEN=-Y_INTERCEPT/SLOPE;
007030             PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ',
007040                      BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ',
007050                      OLD_DATA(6))
007060                     (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
007070          END;
007080       CALL SUM( COOL_ARRAY_OTHER,COOL_OTHER,SUM_X,SUM_Y,SUM_X_Y,
007090                 SUM_X_SQUARED);
007100       OLD_DATA(11)=OLD_DATA(11)+COOL_OTHER;
007110       OLD_DATA(12)=OLD_DATA(12)+SUM_X;
007120       OLD_DATA(13)=OLD_DATA(13)+SUM_Y;
007130       OLD_DATA(14)=OLD_DATA(14)+SUM_X_Y;
007140       OLD_DATA(15)=OLD_DATA(15)+SUM_X_SQUARED;
007150       PUT FILE(SYSPRT3) EDIT ('SOLAR > 150 (BTU/HR-SQFT):')
007160               (SKIP(3),X(5),A);
007170        IF OLD_DATA(11)<2 THEN PUT FILE(SYSPRT3) EDIT
007180          ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
007190          (X(10),A);
007200          ELSE DO;
007210             CALL CURVE(OLD_DATA(12),OLD_DATA(13),OLD_DATA(14),
007220                        OLD_DATA(15),Y_INTERCEPT,SLOPE,OLD_DATA(11));
007230             PUT FILE(SYSPRT3) EDIT
007240                 ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ",
007250                  Y_INTERCEPT)
007260                 (X(10),A,E(12,5,6),A,E(12,5,6));
007270             BREAK_EVEN=-Y_INTERCEPT/SLOPE;
007280             PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ",
007290                      BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ",
007300                      OLD_DATA(11))
007310                     (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
007320          END;
007330       CALL SUM( COOL_ARRAY_ALL,COOL_ALL,SUM_X,SUM_Y,SUM_X_Y,
007340                 SUM_X_SQUARED);
007350       OLD_DATA(31)=OLD_DATA(31)+COOL_ALL;
007360       OLD_DATA(32)=OLD_DATA(32)+SUM_X;
007370       OLD_DATA(33)=OLD_DATA(33)+SUM_Y;
007380       OLD_DATA(34)=OLD_DATA(34)+SUM_X_Y;
007390       OLD_DATA(35)=OLD_DATA(35)+SUM_X_SQUARED;
007400       PUT FILE(SYSPRT3) EDIT (' NO SOLAR DIFFERENTIATION:')
007410               (SKIP(3),X(5),A);
007420        IF OLD_DATA(31)<2 THEN PUT FILE(SYSPRT3) EDIT
007430          ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
007440          (X(10),A);
007450          ELSE DO;
007460             CALL CURVE(OLD_DATA(32),OLD_DATA(33),OLD_DATA(34),
007470                        OLD_DATA(35),Y_INTERCEPT,SLOPE,OLD_DATA(31));
007480             PUT FILE(SYSPRT3) EDIT
007490                 ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
```

```
007500                Y_INTERCEPT)
007510               (X(10),A,E(12,5,6),A,E(12,5,6));
007520            BREAK_EVEN=-Y_INTERCEPT/SLOPE;
007530            PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ',
007540                     BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ',
007550                     OLD_DATA(31))
007560                    (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
007570         END;
007580      HEAD1='***** L E A S T   S Q U A R E D   F I T   F O R   '||
007590            'H E A T I N G   L O A D S *****';
007600      CALL HEADING(HEAD1,PRINT,LINE,LINE_NO);
007610      CALL SUM(HEAT_ARRAY_50,HEAT_50,SUM_X,SUM_Y,SUM_X_Y,
007620               SUM_X_SQUARED);
007630      OLD_DATA(16)=OLD_DATA(16)+HEAT_50;
007640      OLD_DATA(17)=OLD_DATA(17)+SUM_X;
007650      OLD_DATA(18)=OLD_DATA(18)+SUM_Y;
007660      OLD_DATA(19)=OLD_DATA(19)+SUM_X_Y;
007670      OLD_DATA(20)=OLD_DATA(20)+SUM_X_SQUARED;
007680      PUT FILE(SYSPRT3) EDIT ('SOLAR < 50 (BTU/HR-SQFT):')
007690              (SKIP(3),X(5),A);
007700      IF OLD_DATA(16)<2 THEN PUT FILE(SYSPRT3) EDIT
007710         ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
007720         (X(10),A);
007730         ELSE DO;
007740            CALL CURVE(OLD_DATA(17),OLD_DATA(18),OLD_DATA(19),
007750                       OLD_DATA(20),Y_INTERCEPT,SLOPE,OLD_DATA(16));
007760            PUT FILE(SYSPRT3) EDIT
007770                ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
007780                 Y_INTERCEPT)
007790                (X(10),A,E(12,5,6),A,E(12,5,6));
007800            BREAK_EVEN=-Y_INTERCEPT/SLOPE;
007810            PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ',
007820                     BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ',
007830                     OLD_DATA(16))
007840                    (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
007850         END;
007860      CALL SUM(HEAT_ARRAY_150,HEAT_150,SUM_X,SUM_Y,SUM_X_Y,
007870               SUM_X_SQUARED);
007880      OLD_DATA(21)=OLD_DATA(21)+HEAT_150;
007890      OLD_DATA(22)=OLD_DATA(22)+SUM_X;
007900      OLD_DATA(23)=OLD_DATA(23)+SUM_Y;
007910      OLD_DATA(24)=OLD_DATA(24)+SUM_X_Y;
007920      OLD_DATA(25)=OLD_DATA(25)+SUM_X_SQUARED;
007930      PUT FILE(SYSPRT3) EDIT ('SOLAR < 150 (BTU/HR-SQFT):')
007940              (SKIP(3),X(5),A);
007950      IF OLD_DATA(21)<2 THEN PUT FILE(SYSPRT3) EDIT
007960         ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
007970         (X(10),A);
007980         ELSE DO;
007990            CALL CURVE(OLD_DATA(22),OLD_DATA(23),OLD_DATA(24),
008000                       OLD_DATA(25),Y_INTERCEPT,SLOPE,OLD_DATA(21));
008010            PUT FILE(SYSPRT3) EDIT
008020                ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
008030                 Y_INTERCEPT)
008040                (X(10),A,E(12,5,6),A,E(12,5,6));
008050            BREAK_EVEN=-Y_INTERCEPT/SLOPE;
008060            PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ',
008070                     BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ',
```

```
008080                      OLD_DATA(21))
008090                     (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
008100          END;
008110       CALL SUM(HEAT_ARRAY_OTHER,HEAT_OTHER,SUM_X,SUM_Y,SUM_X_Y,
008120                SUM_X_SQUARED);
008130       OLD_DATA(26)=OLD_DATA(26)+HEAT_OTHER;
008140       OLD_DATA(27)=OLD_DATA(27)+SUM_X;
008150       OLD_DATA(28)=OLD_DATA(28)+SUM_Y;
008160       OLD_DATA(29)=OLD_DATA(29)+SUM_X_Y;
008170       OLD_DATA(30)=OLD_DATA(30)+SUM_X_SQUARED;
008180       PUT FILE(SYSPRT3) EDIT ('SOLAR > 150 (BTU/HR-SQFT):')
008190              (SKIP(3),X(5),A);
008200       IF OLD_DATA(26)<2 THEN PUT FILE(SYSPRT3) EDIT
008210          ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
008220          (X(10),A);
008230          ELSE DO;
008240             CALL CURVE(OLD_DATA(27),OLD_DATA(28),OLD_DATA(29),
008250                        OLD_DATA(30),Y_INTERCEPT,SLOPE,OLD_DATA(26));
008260             PUT FILE(SYSPRT3) EDIT
008270                 ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
008280                  Y_INTERCEPT)
008290                 (X(10),A,E(12,5,6),A,E(12,5,6));
008300             BREAK_EVEN=-Y_INTERCEPT/SLOPE;
008310             PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ',
008320                    BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ',
008330                    OLD_DATA(26))
008340                   (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
008350          END;
008360       CALL SUM(HEAT_ARRAY_ALL,HEAT_ALL,SUM_X,SUM_Y,SUM_X_Y,
008370                SUM_X_SQUARED);
008380       OLD_DATA(36)=OLD_DATA(36)+HEAT_ALL;
008390       OLD_DATA(37)=OLD_DATA(37)+SUM_X;
008400       OLD_DATA(38)=OLD_DATA(38)+SUM_Y;
008410       OLD_DATA(39)=OLD_DATA(39)+SUM_X_Y;
008420       OLD_DATA(40)=OLD_DATA(40)+SUM_X_SQUARED;
008430       PUT FILE(SYSPRT3) EDIT (' NO SOLAR DIFFERENTIATION:')
008440              (SKIP(3),X(5),A);
008450       IF OLD_DATA(36)<2 THEN PUT FILE(SYSPRT3) EDIT
008460          ('THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION')
008470          (X(10),A);
008480          ELSE DO;
008490             CALL CURVE(OLD_DATA(37),OLD_DATA(38),OLD_DATA(39),
008500                        OLD_DATA(40),Y_INTERCEPT,SLOPE,OLD_DATA(36));
008510             PUT FILE(SYSPRT3) EDIT
008520                 ('Q (WATTS) = ',SLOPE,' * TEMPERATURE(C) + ',
008530                  Y_INTERCEPT)
008540                 (X(10),A,E(12,5,6),A,E(12,5,6));
008550             BREAK_EVEN=-Y_INTERCEPT/SLOPE;
008560             PUT FILE(SYSPRT3) EDIT ('BREAK EVEN TEMPERATURE = ',
008570                    BREAK_EVEN,' (C)','NO. OF POINTS ANALIZED = ',
008580                    OLD_DATA(36))
008590                   (SKIP,X(41),A,E(12,4,6),A,X(5),A,F(5));
008600          END;
008610       RETURN;
008620  %PAGE;
008630  /*
008640     CURVE - CALCULATES THE SLOPE AND Y_INTERCEPT OF THE CURVE FIT
008650             BASED ON THE SUM'S
```

```
008660  */
008670   CURVE: PROC(SUM_X,SUM_Y,SUM_X_Y,SUM_X_SQUARE,Y_INTERCEPT,SLOPE,#)
008680          REORDER;
008690    DCL (SUM_X,SUM_Y,SUM_X_Y,SUM_X_SQUARE,#) FLOAT DEC(6),
008700        (Y_INTERCEPT,SLOPE) FLOAT BINARY;
008710    (NOOVERFLOW):
008720       SLOPE=(SUM_X_Y-SUM_X*SUM_Y/#)/(SUM_X_SQUARE-SUM_X*SUM_X/#);
008730    (NOOVERFLOW):
008740       Y_INTERCEPT=(SUM_Y-SLOPE*SUM_X)/#;
008750       RETURN;
008760   END CURVE;
008770  %PAGE;
008780  /*
008790     SUM - CALCULATES THE SUM'S NEEDED TO PERFORM THE LEAST SQUARED
008800           FIT FROM THE DATA GIVEN
008810  */
008820   SUM: PROC(ARRAY,#_POINTS,SUM_X,SUM_Y,SUM_X_Y,SUM_X_SQUARE) REORDER;
008830    DCL (I,#_POINTS) FIXED BINARY(15),
008840        (ARRAY(*,*),SUM_X,SUM_Y,SUM_X_Y,
008850         SUM_X_SQUARE) FLOAT BINARY;
008860       SUM_X,SUM_Y,SUM_X_Y,SUM_X_SQUARE=0.;
008870       DO I=1 TO #_POINTS;
008880          SUM_X=SUM_X+ARRAY(2,I); SUM_Y=SUM_Y+ARRAY(1,I);
008890          SUM_X_Y=SUM_X_Y+ARRAY(2,I)*ARRAY(1,I);
008900          SUM_X_SQUARE=SUM_X_SQUARE+ARRAY(2,I)*ARRAY(2,I);
008910       END;
008920       RETURN;
008930   END SUM;
008940   END ENVELOPE;
008950  %PAGE;
008960  /*
008970     ACESSUM - PRODUCES THE LINE PRINTER OUTPUT FOR THE ACES DATA
008980               FILES
008990  */
009000   ACESSUM: PROC REORDER;
009010    DCL (AVG_T,MAX_T,MIN_T,DD_65,DD_60,DD_55,DH_70,DH_75,DH_80,TOT_H,
009020         MAX_H,TOT_C,MAX_C,TOT_E,MAX_E,TOT_W,MAX_W,TOT_E_A,TOT_E_C,
009030         MAX_T_T,MIN_T_T) FLOAT BINARY,
009040        HEAD1 CHAR(79),
009050        (MAX,MIN) BUILTIN,
009060        (DAY,HOUR,#) FIXED BINARY(15);
009070       AVG_T,DD_65,DD_60,DD_55,DH_70,DH_75,DH_80,TOT_H,MAX_H,TOT_C,
009080       MAX_C,TOT_W,MAX_W,TOT_E,MAX_E,TOT_E_A,TOT_E_C,MAX_T_T=0.;
009090       #=0; MIN_T_T=100.;
009100       DO DAY=1 TO 7;
009110          RECORD_#=(DAY-1)*160+100; MIN_T=100.; MAX_T=0.;
009120          READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
009130          DO HOUR=1 TO 24;
009140             IF DATA(HOUR)=99999. THEN GO TO A0;
009150             #=#+1; MAX_T=MAX(DATA(HOUR),MAX_T); AVG_T=AVG_T+DATA(HOUR);
009160             MIN_T=MIN(DATA(HOUR),MIN_T);
009170             IF DATA(HOUR) > 25. THEN DO; DH_80=DH_80+DATA(HOUR)-25.;
009180                DH_75=DH_75+DATA(HOUR)-23.; DH_70=DH_70+DATA(HOUR)-21.;
009190                GO TO A0;
009200             END;
```

```
009210            IF DATA(HOUR) > 23. THEN DO; DH_75=DH_75+DATA(HOUR)-23.;
009220               DH_70=DH_70+DATA(HOUR)-21.; GO TO A0;
009230            END;
009240            IF DATA(HOUR) > 21. THEN DH_70=DH_70+DATA(HOUR)-21.;
009250   A0:    END;
009260          IF (MAX_T+MIN_T)=100. THEN GO TO A1;
009270           IF (MAX_T+MIN_T)/2. < 14. THEN DO;
009280              DD_65=DD_65+18.-(MAX_T+MIN_T)/2.;
009290              DD_60=DD_60+16.-(MAX_T+MIN_T)/2.;
009300              DD_55=DD_55+14.-(MAX_T+MIN_T)/2.; GO TO A1;
009310           END;
009320           IF (MAX_T+MIN_T)/2. < 16. THEN DO;
009330              DD_65=DD_65+18.-(MAX_T+MIN_T)/2.;
009340              DD_60=DD_60+16.-(MAX_T+MIN_T)/2.; GO TO A1;
009350           END;
009360           IF (MAX_T+MIN_T)/2. < 18. THEN
009370              DD_65=DD_65+18.-(MAX_T+MIN_T)/2.;
009380   A1:     RECORD_#=RECORD_#-89; MAX_T_T=MAX(MAX_T,MAX_T_T);
009390           MIN_T_T=MIN(MIN_T,MIN_T_T);
009400           CALL FILL3 (DATA,RECORD_#);
009410           DO HOUR=1 TO 24;
009420              IF DATA(HOUR) = 99999. THEN GO TO A2;
009430              TOT_H=TOT_H+DATA(HOUR); MAX_H=MAX(DATA(HOUR),MAX_H);
009431     /*      MAY 14,1979          */
009432         IF MAX_H >= 10000. THEN MAX_H =0.0;
009440   A2:     END;
009450           RECORD_#=RECORD_#+1;
009460           CALL FILL4 (DATA,RECORD_#);
009470           DO HOUR=1 TO 24;
009480              IF DATA(HOUR) = 99999. THEN GO TO A3;
009490              TOT_C=TOT_C+DATA(HOUR); MAX_C=MAX(DATA(HOUR),MAX_C);
009500   A3:     END;
009510           RECORD_#=RECORD_#+133;
009520           READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
009530           DO HOUR=1 TO 24;
009540              IF DATA(HOUR) = 99999. THEN GO TO A4;
009550              TOT_E=TOT_E+DATA(HOUR); MAX_E=MAX(DATA(HOUR),MAX_E);
009560   A4:     END;
009570           RECORD_#=RECORD_#-132;
009580           READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
009590           DO HOUR=1 TO 24;
009600              IF DATA(HOUR) = 99999. THEN GO TO A5;
009610              TOT_W=TOT_W+DATA(HOUR); MAX_W=MAX(DATA(HOUR),MAX_W);
009620   A5:     END;
009630   /* CHANGED ON 2-5-79 ADD 49,50,51 TO GET TOT_E_C */
009640           RECORD_#=RECORD_#+36;
009650           READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
009660           DO HOUR=1 TO 24;
009670              IF DATA(HOUR) = 99999. THEN GO TO A6;
009680              TOT_E_C=TOT_E_C+DATA(HOUR);
009690   A6:     END;
009700           RECORD_#=RECORD_#+1;
009710           READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
009720           DO HOUR=1 TO 24;
009730              IF DATA(HOUR)=99999. THEN GO TO A7;
009740    /* 2-20-79   REC=50 MULTIPLY BY 2. */
009750    /*        DATA(HOUR)=DATA(HOUR)*2.;          */
009760              TOT_E_C=TOT_E_C+DATA(HOUR);
```

```
009770   A7:     END;
009780           RECORD_#=RECORD_#+1;
009790           READ FILE(ACEDATA) INTO (INPUT_DATA) KEY(RECORD_#);
009800           DO HOUR=1 TO 24;
009810              IF DATA(HOUR) =99999. THEN GO TO A9;
009820              TOT_E_C=TOT_E_C+DATA(HOUR);
009830   A9:     END;
009840           RECORD_#=RECORD_#+90;
009850           READ FILE(ACEDATA) INTO (INPUT_DATA) KEY (RECORD_#);
009860           DO HOUR=1 TO 24;
009870              IF DATA(HOUR) = 99999. THEN GO TO A8;
009880              TOT_E_A=TOT_E_A+DATA(HOUR);
009890   A8:     END;
009900       END;
009910       OLD_DATA(41)=OLD_DATA(41)+#; OLD_DATA(42)=OLD_DATA(42)+AVG_T;
009920       AVG_T=AVG_T/#;
009930       IF (MAX_T_T+MIN_T_T)=100. THEN GO TO AA;
009940       OLD_DATA(43)=MAX(OLD_DATA(43),MAX_T_T);
009950       OLD_DATA(44)=MIN(OLD_DATA(44),MIN_T_T);
009960   AA: OLD_DATA(45)=OLD_DATA(45)+DD_65; OLD_DATA(46)=OLD_DATA(46)+DD_60;
009970       TOT_H=TOT_H/1000.; TOT_C=TOT_C/1000.; TOT_E=TOT_E/1000.;
009980       TOT_W=TOT_W/1000.; TOT_E_C=TOT_E_C/1000.; TOT_E_A=TOT_E_A/1000.;
009990       OLD_DATA(47)=OLD_DATA(47)+DD_55; OLD_DATA(48)=OLD_DATA(48)+DH_70;
010000       OLD_DATA(49)=OLD_DATA(49)+DH_75; OLD_DATA(50)=OLD_DATA(50)+DH_80;
010010       OLD_DATA(51)=OLD_DATA(51)+TOT_H; OLD_DATA(52)=OLD_DATA(52)+TOT_C;
010020       OLD_DATA(53)=OLD_DATA(53)+TOT_E; OLD_DATA(54)=OLD_DATA(54)+TOT_W;
010030       OLD_DATA(55)=OLD_DATA(55)+TOT_E_C;
010040       OLD_DATA(56)=OLD_DATA(56)+TOT_E_A;
010050       OLD_DATA(57)=MAX(OLD_DATA(57),MAX_H);
010060       OLD_DATA(58)=MAX(OLD_DATA(58),MAX_C);
010070       OLD_DATA(59)=MAX(OLD_DATA(59),MAX_E);
010080       OLD_DATA(60)=MAX(OLD_DATA(60),MAX_W);
010090       CALL TITLING(TITLE,PRINT,LINE,PAGE_#); LINE_NO=0;
010100       CALL HEADING(HEAD_DATE,PRINT,LINE,LINE_NO); LINE_NO=3;
010110       HEAD1='***** E Q U I P M E N T   A N D   W E A T H E R   '||
010120            'D A T A   S U M M A R Y *****';
010130       CALL HEADING(HEAD1,PRINT,LINE,LINE_NO); LINE_NO=0;
010140       HEAD1='                        (TO DATE NUMBERS ARE IN PARENTHESES)'
010150            ||'                        ';
010160       CALL HEADING(HEAD1,PRINT,LINE,LINE_NO);
010170       PUT FILE(SYSPRT3) EDIT ('AVERAGE OUTSIDE DRY BULB TEMPERATURE ',
010180                '(C) ----------- ',AVG_T,' (',
010190                OLD_DATA(42)/OLD_DATA(41),')')
010200               (SKIP(5),X(26),A,A,F(12,2),A,F(12,2),A);
010210       PUT FILE(SYSPRT3) EDIT ('MAXIMUM OUTSIDE DRY BULB TEMPERATURE ',
010220                '(C) ----------- ',MAX_T_T,' (',
010230                OLD_DATA(43),')')
010240               (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010250       PUT FILE(SYSPRT3) EDIT ('MINIMUM OUTSIDE DRY BULB TEMPERATURE ',
010260                '(C) ----------- ',MIN_T_T,' (',
010270                OLD_DATA(44),')')
010280               (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010290       PUT FILE(SYSPRT3) EDIT ('DEGREE DAYS HEATING REFERENCED TO 18 ',
010300                '(C) ----------- ',DD_65,' (',
010310                OLD_DATA(45),')')
010320               (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010330       PUT FILE(SYSPRT3) EDIT ('DEGREE DAYS HEATING REFERENCED TO 16 ',
010340                '(C) ----------- ',DD_60,' (',
```

```
010350              OLD_DATA(46),')')
010360             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A );
010370       PUT FILE(SYSPRT3) EDIT ('DEGREE DAYS HEATING REFERENCED TO 14 ',
010380              '(C) ----------- ",DD_55,' ("",
010390              OLD_DATA(47),')')
010400             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A );
010410       PUT FILE(SYSPRT3) EDIT ('DEGREE HOURS COOLING REFERENCED TO 21',
010420              ' (C) ---------- ",DH_70,' ("",
010430              OLD_DATA(48),')')
010440             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010450       PUT FILE(SYSPRT3) EDIT ('DEGREE HOURS COOLING REFERENCED TO 23',
010460              ' (C) ---------- ",DH_75,' ("",
010470              OLD_DATA(49),')')
010480             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010490       PUT FILE(SYSPRT3) EDIT ('DEGREE HOURS COOLING REFERENCED TO 25',
010500              ' (C) ---------- ",DH_80,' ("",
010510              OLD_DATA(50),')')
010520             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A );
010530       PUT FILE(SYSPRT3) EDIT ('TOTAL HEATING LOAD DELIVERED (KWH) ---",
010540              '-------------- ',TOT_H,' ("",
010550              OLD_DATA(51),')')
010560             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A );
010570       PUT FILE(SYSPRT3) EDIT ('TOTAL COOLING LOAD DELIVERED (KWH) ---',
010580              '-------------- ',TOT_C,' ("",
010590              OLD_DATA(52),')')
010600             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010610       PUT FILE(SYSPRT3) EDIT ('TOTAL ECONOMY LOAD DELIVERED (KWH) ---',
010620              '-------------- ",TOT_E,' ("",
010630              OLD_DATA(53),')')
010640             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A );
010650       PUT FILE(SYSPRT3) EDIT ('TOTAL HOT WATER LOAD DELIVERED (KWH) -",
010660              '-------------- ',TOT_W,' ("",
010670              OLD_DATA(54),')')
010680             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010690       PUT FILE(SYSPRT3) EDIT ('MAXIMUM HEATING LOAD DELIVERED (WATT) ",
010700              '-------------- ",MAX_H,' ("",
010710              OLD_DATA(57),')')
010720             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010730       PUT FILE(SYSPRT3) EDIT ('MAXIMUM COOLING LOAD DELIVERED (WATT) "
010740              '-------------- ',MAX_C,' ("",
010750              OLD_DATA(58),')')
010760             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010770       PUT FILE(SYSPRT3) EDIT ('MAXIMUM ECONOMY LOAD DELIVERED (WATT) ",
010780              '-------------- ",MAX_E,' ("",
010790              OLD_DATA(59),')')
010800             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A );
010810       PUT FILE(SYSPRT3) EDIT ('MAXIMUM HOT WATER LOAD DELIVERED (WATT)',
010820              ' ------------ ',MAX_W,' (',
010830              OLD_DATA(60),')')
010840             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010850       PUT FILE(SYSPRT3) EDIT ('TOTAL ACES POWER CONSUMPTION (KWH) ----',
010860              '------------- ',TOT_E_A,' (',
010870              OLD_DATA(56),')')
010880             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
010890      PUT FILE(SYSPRT3) EDIT ('TOTAL CONTROL HOUSE EQUIP POWER CONSUMP',
010900              'TION (KWH) -- ',TOT_E_C,' (',
010910              OLD_DATA(55),')')
010920             (SKIP,X(26),A,A,F(12,2),A,F(12,2),A);
```

```
010930   END ACESSUM;
010940  %PAGE;
010950  /*
010960     WEEK - RETURNS THE WEEK NUMBER FOR THE CURRENT WEEK WITH
010970            10/31/77 REFERENCED AS THE WEEK NUMBER 0
010980  */
010990   WEEK: PROC(EXPERIMENT,ZERO) REORDER;
011000    DCL (YEAR,DAY,EXPERIMENT,ZERO) FIXED BINARY(15),
011010        MONTH(12) FIXED BINARY(15) INIT(0,31,59,90,120,151,181,212,243,
011020                                        273,304,334);
011030      IF WEEK_#>0 THEN GO TO A;
011040      YEAR=IY-77; DAY=MONTH(IM)+ID-304; WEEK_#=(YEAR*365+DAY)/7;
011050      IF WEEK_#=0 THEN DO;
011060         WEEK_#=WEEK_#+1; OLD_DATA(*)=0.; OLD_DATA(44)=100.; RETURN;
011070      END;
011080     IF ZERO=0 THEN DO;
011090     WEEK_#=WEEK_#+1;
011100     OLD_DATA(*)=0.;    OLD_DATA(44)=100.;
011110      RETURN;
011120     END;
011130      READ FILE(TOTAL) INTO (TOTALS) KEY(WEEK_#); WEEK_#=WEEK_#+1;
011140      RETURN;
011150    A: IF EXPERIMENT=1 THEN DO; WEEK_#=WEEK_#-1;
011160         READ FILE(TOTAL) INTO (TOTALS) KEY(WEEK_#); WEEK_#=WEEK_#+1;
011170      END;
011180      WRITE FILE(TOTAL) FROM(TOTALS) KEYFROM(WEEK_#);
011190      CLOSE FILE(TOTAL);
011200      RETURN;
011210   END WEEK;
011220  END ANALIZE;
011230  %PAGE;
011240  END SUMMARY;
011250 * PROCESS;
011260  LOAD: PROC REORDER;
011270           DCL ACEDATA EXT FILE RECORD DIRECT KEYED ENV(REGIONAL(1)),
011280              (I,K,DAY) FIXED BINARY(15),
011290              R_# FIXED DEC(5),
011300              (DATA_51(24),DATA_52(24),DATA_78(24)) FLOAT BINARY,
011310              1 INPUT1,
011320                2 DATA(24) FLOAT DEC(6),
011330              (HEATING(168),COOLING(168),ECONOMY(168),HOT_WATER(168))
011340               STATIC FLOAT BINARY,
011350              Y_ARRAY(168) FLOAT DEC(6);
011360  LOAD1: ENTRY(Y_ARRAY);
011370       GO TO HEAT;
011380  LOAD2: ENTRY(Y_ARRAY);
011390       Y_ARRAY=COOLING; RETURN;
011400  LOAD3: ENTRY(Y_ARRAY);
011410       Y_ARRAY=HOT_WATER; RETURN;
011420  LOAD4: ENTRY(Y_ARRAY);
011430       Y_ARRAY=ECONOMY; RETURN;
011440  LOAD5: ENTRY(Y_ARRAY);
011450       Y_ARRAY(*)=0.;
011460       DO I=1 TO 168;
011470          IF HEATING(I)=99999. THEN GO TO Z1;
011480          Y_ARRAY(I)=HEATING(I);
011490  Z1:     IF COOLING(I)=99999. THEN GO TO Z2;
011500          Y_ARRAY(I)=Y_ARRAY(I)+COOLING(I);
```

```
011510  Z2:     IF HOT_WATER(I)=99999. THEN GO TO Z3;
011520          Y_ARRAY(I)=Y_ARRAY(I)+HOT_WATER(I);
011530  Z3:     IF ECONOMY(I)=99999. THEN GO TO Z4;
011540          Y_ARRAY(I)=Y_ARRAY(I)+ECONOMY(I);
011550  Z4:     IF Y_ARRAY(I)>99999. THEN Y_ARRAY(I)=99999.;
011560          IF Y_ARRAY(I)<0. THEN Y_ARRAY(I)=0.;
011570      END; RETURN;
011580  HEAT:
011590      DO DAY=1 TO 7;
011600         R_#=(DAY-1)*160+11;
011610         CALL FILLA1(DATA_51,R_#); CALL FILLA2(DATA_52,R_#);
011620         R_#=R_#+134; READ FILE( ACEDATA ) INTO (INPUT1) KEY(R_#);
011630         DATA_78=DATA; R_#=R_#-132;
011640         READ FILE( ACEDATA ) INTO (INPUT1) KEY(R_#); K=(DAY-1)*24;
011650         DO I=1 TO 24;
011660            K=K+1; COOLING(K)=DATA_52(I); HOT_WATER(K)=DATA(I);
011670            HEATING(K)=DATA_51(I); ECONOMY(K)=DATA_78(I);
011680            IF HEATING(K)<0. THEN HEATING(K)=0.;
011690            IF HEATING(K)>99999. THEN HEATING(K)=99999.;
011700      END; END; Y_ARRAY=HEATING; RETURN;
011710  END LOAD;
011720  * PROCESS;
011730  FILL1: PROC (DATA1,R_#) REORDER;
011740          DCL (DATA1(24),DATA_41(24),DATA_42(24),DATA_43(24),DATA_44(24),
011750               DATA_55(24),DATA_45(24),DATA_46(24),DATA_47(24),
011760               DATA_48(24),DATA_49(24)) FLOAT BINARY,
011770              DATA2(24) FLOAT DEC(6),
011780              (DATA_52(24),DATA_53(24)) STATIC FLOAT BINARY,
011790              (I,J) FIXED BINARY(15),
011800              R_# FIXED DEC(5),
011810              1 INPUT1,
011820                2 DATA(24) FLOAT DEC(6),
011830              ACEDATA EXT FILE RECORD DIRECT KEYED ENV(REGIONAL(1));
011840      I=1; GO TO HEAT;
011850  FILL2: ENTRY (DATA1,R_#);
011860      I=1; GO TO COOL;
011870  FILL3: ENTRY (DATA2,R_#);
011880      I=2; GO TO HEAT;
011890  FILL4: ENTRY (DATA2,R_#);
011900      I=2;
011910  COOL:
011920      IF I=1 THEN GO TO COOL_1;
011930      DATA2=DATA_52; RETURN;
011940  COOL_1:
011950      DATA1=DATA_52; RETURN;
011960  HEAT:
011970      READ FILE( ACEDATA ) INTO (INPUT1) KEY (R_#);
011980      IF I=1 THEN DATA1=DATA; ELSE DATA2=DATA; R_#=R_#-9;
011990      READ FILE( ACEDATA ) INTO (INPUT1) KEY (R_#);
012000      DATA_41=DATA; R_#=R_#-1;
012010      READ FILE( ACEDATA ) INTO (INPUT1) KEY (R_#);
012020      DATA_42=DATA; R_#=R_#+3;
012030      READ FILE( ACEDATA ) INTO (INPUT1) KEY (R_#);
012040      DATA_43=DATA; R_#=R_#-1;
012050      READ FILE( ACEDATA ) INTO (INPUT1) KEY (R_#);
012060      DATA_44=DATA; R_#=R_#+2;
012070      READ FILE( ACEDATA ) INTO (INPUT1) KEY (R_#);
012080      DATA_45=DATA; R_#=R_#+1;
```

```
012090       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012100       DATA_46=DATA; R_#=R_#+1;
012110       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012120       DATA_47=DATA; R_#=R_#+1;
012130       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012140       DATA_48=DATA; R_#=R_#+1;
012150       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012160       DATA_49=DATA; R_#=R_#+3;
012170       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012180       DATA_52=DATA; R_#=R_#+1;
012190       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012200       DATA_53=DATA; R_#=R_#+128;
012210       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012220       DATA_55=DATA; R_#=R_#-128;
012230       READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#); R_#=R_#-2;
012240       IF I=1 THEN GO TO HEAT_1;
012250       DO J=1 TO 24;
012260          DATA2(J)=DATA_43(J)-DATA_44(J)+DATA_41(J)-DATA_42(J)
012270                   +DATA_55(J)-DATA(J);
012280          IF DATA_52(J)=0. THEN GO TO Z3;
012290          IF DATA_52(J)=99999. THEN GO TO Z3;
012300          DATA_45(J)=DATA_45(J)+DATA_46(J)+DATA_47(J);
012310          IF DATA_45(J)>=99999. THEN DATA_45(J)=0.;
012320          IF DATA_48(J)=99999. THEN DATA_48(J)=0.;
012330          IF DATA_49(J)=99999. THEN DATA_49(J)=0.;
012340          DATA_45(J)=0.75*DATA_45(J)+0.5*DATA_48(J)+DATA_49(J);
012350          DATA_52(J)=DATA_52(J)-DATA_45(J);
012360          DATA2(J)=0.;
012370       Z3: IF DATA2(J)<0. THEN DATA2(J)=0.;
012380          IF DATA2(J)>99999. THEN DATA2(J)=99999.;
012390          IF DATA_52(J)>99999. THEN DATA_52(J)=99999.;
012400          IF DATA_52(J)<0. THEN DATA_52(J)=0.;
012410       END; RETURN;
012420  HEAT_1:
012430       DO J=1 TO 24;
012440          DATA1(J)=DATA_43(J)-DATA_44(J)+DATA_41(J)-DATA_42(J)
012450                   +DATA_55(J)-DATA(J);
012460          IF DATA_52(J)=0. THEN GO TO Z4;
012470          IF DATA_52(J)=99999. THEN GO TO Z4;
012480          DATA_45(J)=DATA_45(J)+DATA_46(J)+DATA_47(J);
012490          IF DATA_45(J)>=99999. THEN DATA_45(J)=0.;
012500          IF DATA_48(J)=99999. THEN DATA_48(J)=0.;
012510          IF DATA_49(J)=99999. THEN DATA_49(J)=0.;
012520          DATA_45(J)=0.75*DATA_45(J)+0.5*DATA_48(J)+DATA_49(J);
012530          DATA_52(J)=DATA_52(J)-DATA_45(J);
012540          DATA1(J)=0.;
012550       Z4: IF DATA1(J)<0. THEN DATA1(J)=0.;
012560          IF DATA1(J)>99999. THEN DATA1(J)=99999.;
012570          IF DATA_52(J)>99999. THEN DATA_52(J)=99999.;
012580          IF DATA_52(J)<0. THEN DATA_52(J)=0.;
012590       END; RETURN;
012600       END FILL1;
012610  * PROCESS;
012620  FILLA1: PROC (DATA1,R_#) REORDER;
012630          DCL (DATA1(24),DATA_41(24),DATA_42(24),DATA_43(24),DATA_44(24),
012640               DATA_55(24),DATA_45(24),DATA_46(24),DATA_47(24),
012650               DATA_48(24),DATA_49(24)) :LOAT BINARY,
012660              DATA2(24) :LOAT DEC(6),
```

```
012670              (DATA_52(24),DATA_53(24)) STATIC FLOAT BINARY,
012680              (I,J) FIXED BINARY(15),
012690              R_# FIXED DEC(5),
012700              1 INPUT1,
012710                2 DATA(24) FLOAT DEC(6),
012720              ACEDATA EXT FILE RECORD DIRECT KEYED ENV(REGIONAL(1));
012730        I=1; GO TO HEAT;
012740   FILLA2: ENTRY (DATA1,R_#);
012750        I=1; GO TO COOL;
012760   FILLA3: ENTRY (DATA2,R_#);
012770        I=2; GO TO HEAT;
012780   FILLA4: ENTRY (DATA2,R_#);
012790        I=2;
012800   COOL:
012810        IF I=1 THEN GO TO COOL_1;
012820        DATA2=DATA_52; RETURN;
012830   COOL_1:
012840        DATA1=DATA_52; RETURN;
012850   HEAT:
012860        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012870        IF I=1 THEN DATA1=DATA; ELSE DATA2=DATA; R_#=R_#-9;
012880        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012890        DATA_41=DATA; R_#=R_#-1;
012900        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012910        DATA_42=DATA; R_#=R_#+3;
012920        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012930        DATA_43=DATA; R_#=R_#-1;
012940        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012950        DATA_44=DATA; R_#=R_#+2;
012960        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012970        DATA_45=DATA; R_#=R_#+1;
012980        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
012990        DATA_46=DATA; R_#=R_#+1;
013000        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
013010        DATA_47=DATA; R_#=R_#+1;
013020        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
013030        DATA_48=DATA; R_#=R_#+1;
013040        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
013050        DATA_49=DATA; R_#=R_#+3;
013060        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
013070        DATA_52=DATA; R_#=R_#+1;
013080        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
013090        DATA_53=DATA; R_#=R_#+128;
013100        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#);
013110        DATA_55=DATA; R_#=R_#-128;
013120        READ FILE(ACEDATA) INTO (INPUT1) KEY (R_#); R_#=R_#-2;
013130        IF I=1 THEN GO TO HEAT_1;
013140        DO J=1 TO 24;
013150           DATA2(J)=DATA_43(J)-DATA_44(J)+DATA_41(J)-DATA_42(J)
013160                    +DATA_55(J)-DATA(J);
013170           IF DATA_52(J)=0. THEN GO TO Z3;
013180           IF DATA_52(J)=99999. THEN GO TO Z3;
013190           DATA_45(J)=DATA_45(J)+DATA_46(J)+DATA_47(J);
013200           IF DATA_45(J)>=99999. THEN DATA_45(J)=0.;
013210           IF DATA_48(J)=99999. THEN DATA_48(J)=0.;
013220           IF DATA_49(J)=99999. THEN DATA_49(J)=0.;
013230           DATA_45(J)=0.75*DATA_45(J)+0.5*DATA_48(J)+DATA_49(J);
013240           DATA_52(J)=DATA_52(J)-DATA_45(J);
```

```
013250          DATA2(J)=0.;
013260       Z3: IF DATA2(J)<0. THEN DATA2(J)=0.;
013270          IF DATA2(J)>99999. THEN DATA2(J)=99999.;
013280          IF DATA_52(J)>99999. THEN DATA_52(J)=99999.;
013290          IF DATA_52(J)<0. THEN DATA_52(J)=0.;
013300        END; RETURN;
013310  HEAT_1:
013320        DO J=1 TO 24;
013330          DATA1(J)=DATA_43(J)-DATA_44(J)+DATA_41(J)-DATA_42(J)
013340                   +DATA_55(J)-DATA(J);
013350          IF DATA_52(J)=0. THEN GO TO Z4;
013360          IF DATA_52(J)=99999. THEN GO TO Z4;
013370          DATA_45(J)=DATA_45(J)+DATA_46(J)+DATA_47(J);
013380          IF DATA_45(J)>=99999. THEN DATA_45(J)=0.;
013390          IF DATA_48(J)=99999. THEN DATA_48(J)=0.;
013400          IF DATA_49(J)=99999. THEN DATA_49(J)=0.;
013410          DATA_45(J)=0.75*DATA_45(J)+0.5*DATA_48(J)+DATA_49(J);
013420          DATA_52(J)=DATA_52(J)-DATA_45(J);
013430          DATA1(J)=0.;
013440       Z4: IF DATA1(J)<0. THEN DATA1(J)=0.;
013450          IF DATA1(J)>99999. THEN DATA1(J)=99999.;
013460          IF DATA_52(J)>99999. THEN DATA_52(J)=99999.;
013470          IF DATA_52(J)<0. THEN DATA_52(J)=0.;
013480        END; RETURN;
013490        END FILLA1;
013491 * PROCESS;
013500  TITLING: PROC (TITLE,PRINT_FILE,LINE,PAGE_#)
013510            REORDER OPTIONS(TASK,REENTRANT);
013520     DCL (L,PAGE_#,LINE,LINE_NO) FIXED BINARY(15),
013530          PRINT_FILE FILE VARIABLE,
013540          TITLE CHAR(*),
013550          LENGTH BUILTIN;
013560        L=LENGTH(TITLE); L=(LINE-L)/2; PAGE_#=PAGE_#+1;
013570        PUT FILE(PRINT_FILE) EDIT (TITLE,'PAGE',PAGE_#,'')
013580                (PAGE,X(L),A,X(L-9),A,F(5),SKIP,A);
013590        RETURN;
013600  HEADING: ENTRY (TITLE,PRINT_FILE,LINE,LINE_NO);
013610        L=LENGTH(TITLE); L=(LINE-L)/2;
013620        PUT FILE(PRINT_FILE) EDIT (TITLE,'')
013630                (SKIP(LINE_NO),X(L),A,SKIP,A);
013640        RETURN;
013650  END TITLING;
END OF DATA
READY
```

Appendix A.4

MONTHLY

```
DSNAME='IJL.MONTHLY.FORT'
 C234567890
       REAL*8 MONTH
       DIMENSION MONTH(12),HEAT(24),COOL(24),HW(24),OD(24),HR(24),SOL(24)
       DIMENSION NDAY(12)
       DATA NDAY /31,28,31,30,31,30,31,31,30,31,30,31/
       DATA MONTH /8H JANUARY,8HFEBRUARY,8H  MARCH ,8H  APRIL ,
      >            8H   MAY  ,8H  JUNE  ,8H  JULY  ,8H AUGUST ,
      >            8HSEPTEMBR,8H OCTOBER,8HNOVEMBER,8HDECEMBER/
       REWIND 10
       SOLY=0.
       DDY=0.
       HTY=0.
       CLY=0.
       HWY=0.
       AVGTY=0.
       HR80Y=0.
       HR65Y=0.
       HR39Y=0.
       HR35Y=0.
       HR25Y=0.
       HR10Y=0.
       PRINT 10
    10 FORMAT(' ENTER THE FIRST MONTH TO BE PRINTED')
       READ(5,11) IST
    11 FORMAT(I2)
       PRINT 12
    12 FORMAT(' ENTER THE LAST MONTH TO BE PRINTED')
       READ(5,11) ILT
       PRINT 13
    13 FORMAT(' DO YOU WANT AN ECONOMY CYCLE',/,
      >       '   (0=NO/1=TEMPERATURE CONTROL/2=HUMIDITY CONTROL)')
       READ(5,14) IECON
    14 FORMAT(I1)
       DO 1000 IM=1,12
       SOLM=0.
       DDM=0.
       HTM=0.
       CLM=0.
       HWM=0.
       AVGTM=0.
       TMAX=-200.
       TMIN=200.
       HR80M=0.
       HR65M=0.
       HR39M=0.
       HR35M=0.
       HR25M=0.
       HR10M=0.
       ND=NDAY(IM)
       DO 2000 ID=1,ND
       READ(10) HEAT,COOL,HW,OD,HR,SOL
       DO 3000 IHR=1,24
```

```
      COOL(IHR)=COOL(IHR)*3.412
      HEAT(IHR)=HEAT(IHR)*3.412
      HW(IHR)=HW(IHR)*3.412
      IF (COOL(IHR) .LE. 0.) HEAT(IHR)=HEAT(IHR)*1.2+HW(IHR)*0.2
      IF (COOL(IHR) .GT. 0.) COOL(IHR)=COOL(IHR)*0.95
      IF (IECON .GE. 1 .AND. COOL(IHR) .GT. 0)
     >   CALL ECON(COOL(IHR),OD(IHR),HR(IHR),IECON)
      IF (HW(IHR) .GT. 0.) HW(IHR)=HW(IHR)
      IF (SOL(IHR) .GT. 0.) SOL(IHR)=SOL(IHR)/3.15472
      OD(IHR)=OD(IHR)*9./5.+32.
      IF (OD(IHR) .LE. 10.) GO TO 100
      IF (OD(IHR) .LT. 25.) GO TO 200
      IF (OD(IHR) .LT. 35.) GO TO 300
      IF (OD(IHR) .LT. 39.) GO TO 400
      IF (OD(IHR) .LT. 65.) GO TO 450
      IF (OD(IHR) .LT. 80.) GO TO 500
      GO TO 600
  100 HR10M=HR10M+1
  200 HR25M=HR25M+1
  300 HR35M=HR35M+1
  400 HR39M=HR39M+1
  450 HR65M=HR65M+1
  500 HR80M=HR80M+1
  600 HTM=HEAT(IHR)+HTM
      CLM=COOL(IHR)+CLM
      HWM=HW(IHR)+HWM
      SOLM=SOL(IHR)+SOLM
      TMAX=AMAX1(OD(IHR),TMAX)
      TMIN=AMIN1(OD(IHR),TMIN)
      AVGTM=AVGTM+OD(IHR)
 3000 CONTINUE
C     IF ((TMAX+TMIN)/2. .LT. 65.) DDM=DDM-(TMAX+TMIN)/2.+65.
C     IF( AVGTM .LT. 65.) DDM=DDM-AVGTM+65.
 2000 CONTINUE
      SOLY=SOLY+SOLM
C     DDY=DDY+DDM
      HTY=HTY+HTM
      CLY=CLY+CLM
      HWY=HWY+HWM
      AVGTY=AVGTY+AVGTM
      HR80Y=HR80Y+HR80M
      HR65Y=HR65Y+HR65M
      HR39Y=HR39Y+HR39M
      HR35Y=HR35Y+HR35M
      HR25Y=HR25Y+HR25M
      HR10Y=HR10Y+HR10M
      AVGTM=AVGTM/24./FLOAT(NDAY(IM))
      DDM=DDM-((AVGTM-65.)*HR65M/24.)
      IF(DDM .LT. 0.0) DDM=0.0
      DDY=DDY+DDM
      IF (IM .LT. IST) GO TO 1000
      IF (IM .GT. ILT) GO TO 1000
      PRINT 9000,MONTH(IM)
 9000 FORMAT(//,36X,A8,//,1X,80(1H*),/)
      PRINT 9010,HTM,CLM,HWM,SOLM
 9010 FORMAT(1H ,'TOTAL MONTHLY HEATING LOAD (BTU) --------------',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY COOLING LOAD (BTU) --------------',
```

```
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU) ---',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT) -----',
     >       19(1H-),F12.2)
             PRINT 9020,DDM,AVGTM,HR80M,HR65M
9020 FORMAT(1H ,'TOTAL MONTHLY DEGREE DAYS HEATING -------------',
     >       19(1H-),F12.2,/,
     >       1H ,'AVERAGE MONTHLY DRY BULB TEMPERATURE (F) ------',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY HOURS LESS THAN 80 DEG F --------',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY HOURS LESS THAN 65 DEG F --------',
     >       19(1H-),F12.2)
      PRINT 9030,HR39M,HR35M,HR25M,HR10M
9030 FORMAT(1H ,'TOTAL MONTHLY HOURS LESS THAN 39 DEG F --------',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY HOURS LESS THAN 35 DEG F --------',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY HOURS LESS THAN 25 DEG F --------',
     >       19(1H-),F12.2,/,
     >       1H ,'TOTAL MONTHLY HOURS LESS THAN 10 DEG F --------',
     >       19(1H-),F12.2,//,
     >       1H ,80(1H*),////)
      PRINT 8000
      READ(5,8010) A
1000 CONTINUE
      AVGTY=AVGTY/24./365.
      PRINT 9040
9040 FORMAT(//,36X,'  YEAR  ',//,1X,80(1H*),/)
      HTY=HTY/1000000.
      CLY=CLY/1000000.
      HWY=HWY/1000000.
      SOLY=SOLY/1000.
      PRINT 9050,HTY,CLY,HWY,SOLY
9050 FORMAT(1H ,'TOTAL YEARLY HEATING LOAD (MBTU) -------------------',
     >       19(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY COOLING LOAD (MBTU) -------------------',
     >       19(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY DOMESTIC HOT WATER LOAD (MBTU) -------',
     >       19(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY SOLAR RADIATION (TBTU/FT/FT) ---------',
     >       19(1H-),F10.2)
      PRINT 9060,DDY,AVGTY,HR80Y,HR65Y
9060 FORMAT(1H ,'TOTAL YEARLY DEGREE DAYS HEATING -----------------',
     >       20(1H-),F10.2,/,
     >       1H ,'AVERAGE YEARLY DRY BULB TEMPERATURE (F) ----------',
     >       20(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY HOURS LESS THAN 80 DEG F ------------',
     >       20(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY HOURS LESS THAN 65 DEG F ------------',
     >       20(1H-),F10.2)
```

```
      PRINT 9070,HR39Y,HR35Y,HR25Y,HR10Y
 9070 FORMAT(1H ,'TOTAL YEARLY HOURS LESS THAN 39 DEG F ----------',
     >       20(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY HOURS LESS THAN 35 DEG F ----------',
     >       20(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY HOURS LESS THAN 25 DEG F ----------',
     >       20(1H-),F10.2,/,
     >       1H ,'TOTAL YEARLY HOURS LESS THAN 10 DEG F ----------',
     >       20(1H-),F10.2,//,
     >       1H ,80(1H*),////)
      PRINT 8000
 8000 FORMAT(/////,' HIT - RETURN AFTER TEARING OFF THE PAPER',////)
      READ(5,8010) A
 8010 FORMAT(A4)
      STOP
      END
      SUBROUTINE ECON (COOL,OD,HR,IECON)
      T=OD*9./5.+32.
      IF (IECON .EQ. 2) GO TO 10
      IF (T .LT. 72.) COOL=0.
      GO TO 20
   10 ENTHAL=0.24*T+HR*(1061.+0.444*T)
      IF (ENTHAL .GE. 32.20) GO TO 20
      Q=(32.2-ENTHAL)*800*60/13.5
      COOL=COOL-Q
      IF (COOL .LT. 0.) COOL=0.
   20 RETURN
      END
END OF DATA
READY
```

Appendix A.5

SOLAR

```
DSNAME='IJL.SOLAR.FORT'
000010 C
000020 C  PROGRAM TO PRODUCE SOLAR SUMMARY
000030 C
000040          REAL*8 DATE
000050          DIMENSION SUM(10),R(9,24,7)
000060 C  R(1,J,K)=RECORD 114
000070 C  R(2,J,K)=RECORD 31
000080 C  R(3,J,K)=RECORD 32
000090 C  R(4,J,K)=RECORD 34
000100 C  R(5,J,K)=RECORD 36
000110 C  R(6,J,K)=RECORD 37
000120 C  R(7,J,K)=RECORD 38
000130 C  R(8,J,K)=RECORD 39
000140 C  R(9,J,K)=RECORD 40
000150 C  R(10,J,K)=RECORD 33
000160 C  R(11,J,K)=RECORD 35
000170 C  R(12,J,K)=RECORD 41
000180 C  R(13,J,K)=RECORD 42
000190 C  READ IN DATA
000200          READ(1,105) DATE
000210 105      FORMAT(A8)
000220          DO 20 K=1,7
000230          DO 20 I=1,9
000240          READ(1,100) (R(I,J,K),J=1,24)
000250 100      FORMAT(8F10.0)
000260 20       CONTINUE
000270 C
000280 C  ZERO SUM
000290 C
000300          DO 30 I=1,9
000310 30       SUM(I)=0.0
000320 C
000330 C  WRITE HEADING
000340 C
000350          PRINT 110,DATE
000360 110      FORMAT('1',T14,'***** S O L A R  E N E R G Y  S Y S T E M',
000370      >          '  D E M O N S T R A T I O N  P R O J E C T *****',
000380      >          T125,'PAGE  4'/1X,T47,'ANALYZED FOR WEEK BEGINNING - ',
000390      >          A8///1X,T24,'***** E Q U I P M E N T  A N D  W E A T',
000400      >          'H E R  D A T A  S U M M A R Y *****'////)
000410 C
000420 C  CALCULATE AND WRITE DATA
000430 C  ITEMS 1,2,3
000440 C
000450          DO 40 K=1,7
000460          DO 40 J=1,24
000470          IF(R(2,J,K).NE.99999.) SUM(1)=SUM(1)+R(2,J,K)
000480          IF(R(3,J,K).NE.99999.) SUM(2)=SUM(2)+R(3,J,K)
000500          IF(R(8,J,K).NE.99999.) SUM(3)=SUM(4)+R(8,J,K)
000510 40       CONTINUE
000520          SUM(1)=SUM(1)/1000.
000530          SUM(2)=SUM(2)/1000.
000540          SUM(3)=SUM(3)/1000.
```

```
000560          PRINT 120, SUM(1)
000570 120      FORMAT(1X,T29,'TOTAL SOLAR ENERGY COLLECTED',T83,F10.2,'  KWH')
000580          PRINT 130, SUM(2)
000590 130      FORMAT(1X,T29,'SOLAR SPACE HEATING ENERGY DELIVERED',T83,
000600       >  F10.2,'  KWH')
000610         PRINT 140,SUM(3)
000660 140      FORMAT(1X,T29,'ELECTRIC SPACE HEATING ENERGY DELIVERED',T83,
000670       >  F10.2,'  KWH')
000680 C
000690 C  ITEM 4
000700 C
000710          SUM(4)=SUM(2)+SUM(3)
000720          PRINT 150, SUM(4)
000730 150      FORMAT(1X,T29,'TOTAL SPACE HEATING ENERGY DELIVERED',T83,
000740       >  F10.2,'  KWH')
000750 C
000760 C  ITEMS 5 AND 6
000770 C
000780          DO 50 K=1,7
000790          DO 50 J=1,24
000800          IF(R(4,J,K).NE.99999.) SUM(5)=SUM(5)+R(4,J,K)
000830          IF(R(6,J,K).NE.99999.) SUM(6)=SUM(6)+R(6,J,K)
000840 50       CONTINUE
000850          SUM(5)=SUM(5)/1000.
000860          SUM(6)=SUM(6)/1000.
000870          PRINT 160, SUM(5)
000880 160      FORMAT(1X,T29,'SOLAR WATER HEATING ENERGY DELIVERED',T83,
000890       >  F10.2,'  KWH')
000900          PRINT 170, SUM(7)
000910 170      FORMAT(1X,T29,'ELECTRIC WATER HEATING ENERGY DELIVERED',T83,
000920       >  F10.2,'  KWH')
000930 C
000940 C  ITEM 7
000950 C
000960          SUM(6)=SUM(5)+SUM(7)
000970          PRINT 180, SUM(7)
000980 180      FORMAT(1X,T29,'TOTAL WATER HEATING ENERGY DELIVERED',T83,
000990       >  F10.2,'  KWH')
001000 C
001010 C  ITEM 8
001020 C
001030          DO 60 K=1,7
001040          DO 60 J=1,24
001050          IF(R(5,J,K).NE.99999.)  SUM(8)=SUM(8)+R(5,J,K)
001060        IF(R(7,J,K).NE.99999.) SUM(8)=SUM(8)+R(7,J,K)
001061         IF(R(9,J,K).NE.99999.) SUM(8)=SUM(8)+R(9,J,K)
001090 60        CONTINUE
001100          SUM(8)=SUM(8)/1000.
001130          PRINT 190, SUM(8)
001131  190     FORMAT(1X,T29,'TOTAL SOLAR SYSTEM ELECTRICITY',T83,
001132       >  'CONSUMPTION',T83,F10.2,'  KWH')
001160 C
```

```
001170 C  ITEM 9
001180 C
001220 C
001290 200     FORMAT(1X,T29,'TOTAL HEATING AND HOT WATER ELECTRICITY ',
001300       >  'CONSUMPTION',T83,F10.2,'  KWH')
001301             SUM(9)=SUM(8)+SUM(3)+SUM(6)
001360          PRINT 200, SUM(9)
001390 C
001391 C  ITEM 10
001410          DO 70 J=1,24
001420          DO 70 K=1,7
001430          X=AMIN1(X,R(1,J,K))
001440 70       CONTINUE
001450          SUM(10)=X
001460          PRINT 210, SUM(10)
001470 210      FORMAT(1X,T29,'MINIMUM INDOOR TEMPERATURE FOR SOLAR HOUSE',T83,
001480       >   F10.2,'   C')
001490          STOP
001500          END
END OF DATA
READY
```

Appendix A.6

PLOT

```
DSNAME='IJL.PLOTTER.FORT'
                    INTEGER PLOT
PLOT    2         DIMENSION YARRAY(168),XARRAY(168),DATE(20)
PLOT    3         REWIND 10
PLOT    4         CALL FIXUP
PLOT    5         READ (10,100) DATE
PLOT    6     100 FORMAT (20A4)
PLOT    7         CALL PLTPRG
PLOT    8         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT    9         PLOT=1
PLOT   10         CALL PLTNO1 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   11         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT   12         PLOT=2
PLOT   13         CALL PLTNO1 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   14         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT   15         PLOT=3
PLOT   16         CALL PLTNO1 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   17         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT   18         PLOT=4
PLOT   19         CALL PLTNO1 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   20         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT   21         PLOT=5
PLOT   22         CALL PLTNO1 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   23         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   24         PLOT=1
PLOT   25         CALL PLTNO2 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   26         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   27         PLOT=2
PLOT   28         CALL PLTNO2 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   29         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   30         PLOT=3
PLOT   31         CALL PLTNO2 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   32         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   33         PLOT=4
PLOT   34         CALL PLTNO2 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   35         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   36         PLOT=5
PLOT   37         CALL PLTNO2 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   38         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT   39         PLOT=1
PLOT   40         CALL PLTNO3 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   41         CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT   42         PLOT=2
PLOT   43         CALL PLTNO3 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   44         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   45         PLOT=3
PLOT   46         CALL PLTNO3 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   47         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   48         PLOT=1
PLOT   49         CALL PLTNO4 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT   50         CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT   51         PLOT=2
```

```
PLOT  52        CALL PLTNO4 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  53        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  54        PLOT=3
PLOT  55        CALL PLTNO4 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  56        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  57        PLOT=4
PLOT  58        CALL PLTNO4 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  59        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  60        PLOT=5
PLOT  61        CALL PLTNO4 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  62        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  63        PLOT=1
PLOT  64        CALL PLTNO5 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  65        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  66        PLOT=2
PLOT  67        CALL PLTNO5 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  68        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  69        PLOT=3
PLOT  70        CALL PLTNO5 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  71        CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT  72        PLOT=1
PLOT  73        CALL PLTNO6 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  74        CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT  75        PLOT=2
PLOT  76        CALL PLTNO6 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  77        CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT  78        PLOT=3
PLOT  79        CALL PLTNO6 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  80        CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT  81        PLOT=4
PLOT  82        CALL PLTNO6 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  83        CALL FIXUP1 (XARRAY,YARRAY,NOPTS)
PLOT  84        PLOT=5
PLOT  85        CALL PLTNO6 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  86        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  87        PLOT=1
PLOT  88        CALL PLTNO7 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  89        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  90        PLOT=2
PLOT  91        CALL PLTNO7 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT  92        CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
PLOT  93        PLOT=1
PLOT  94        CALL PLTNO8 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=2
              CALL PLTNO8 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=3
              CALL PLTNO8 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=1
              CALL PLTNO9 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=2
              CALL PLTNO9 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=3
```

```
              CALL PLTNO9 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=4
              CALL PLTNO9 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
              CALL FIXUP0 (XARRAY,YARRAY,NOPTS)
              PLOT=5
              CALL PLTNO9 (XARRAY,YARRAY,NOPTS,DATE(2),PLOT)
PLOT 101        CALL ENDPLT
OT 102          STOP
PLOT 103        END
PLOT 104        SUBROUTINE PLTPRG
PLOT 105 C
PLOT 106 C    ************************************************************************
*
PLOT 107 C    ******************************        ******************************
*
PLOT 108 C    ****************************** PLTPRG ******************************
*
PLOT 109 C    ******************************        ******************************
*
PLOT 110 C    ************************************************************************
*
PLOT 111 C    *
*
PLOT 112 C    *     SUBROUTINE - PLTPRG PLOTS THE DATA FOR THE ACES DATA REDUC-
*
PLOT 113 C    *     TION PROGRAM.  THE PLOTS RETURNED WILL BE IN EITHER METRIC OR
*
PLOT 114 C    *     ENGLISH UNITS.  TO OBTAIN ENGLISH UNITS, PUT A 0 IN COLUMN 1
*
PLOT 115 C    *     OF THE FIRST CARD ON FILE WITH DDNAME FT05F001.  ANY OTHER
*
PLOT 116 C    *     NUMBER WILL PRODUCE METRIC UNITS ON THE PLOTS.
*
PLOT 117 C    *

PLOT 118 C    *     ANY DIFFICULTIES SHOULD BE REPORTED TO VAN BAXTER
*
PLOT 119 C    *     BLDG 9102-1 RM 22.
*
PLOT 120 C    *

PLOT 121 C    ************************************************************************
*
PLOT 122 C
PLOT 123        INTEGER PLOT,ONE,TWO,THREE,FOUR,FIVE
PLOT 124        DIMENSION DAYS(16),XARRAY(NOPTS),YARRAY(NOPTS),DATE(7),LEGND(70)
PLOT 125        DATA ONE,TWO,THREE,FOUR,FIVE/1,2,3,4,5/
PLOT 126        DATA DAYS/4H(MON,4H)    ,4H(TUE,4H)    ,4H(WED,4H)    ,4H(THU,4H)
,
PLOT 127       >          4H(FRI,4H)    ,4H(SAT,4H)    ,4H(SUN,4H)    ,4H(MON,4H)
/
PLOT 128 C
PLOT 129 C--- SUBROUTINE COMPRS SETS UP  THE PLOTS FOR USE BY THE DISSPLA
PLOT 130 C--- POST PROCESSOR.
PLOT 131 C
              CALL CALCMP
PLOT 133       RETURN
```

```
      ENTRY PLTNO1 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
C
C   ************************************************************************
*
C   ******************************        ******************************
*
C   ****************************** PLTNO1 ******************************
*
C   ******************************        ******************************
*
C   ************************************************************************
*
C   *
*
C   *    ENTRY - PLTNO1 PRODUCES THE FIRST PLOT OF THE ACES DATA
*
C   *    THIS PLOT DRAWS THE FOLLOWING 5 CURVES.
*
C   *
*
C   *    CURVE 1 - OUTSIDE AIR TEMPERATURE (DB) IN <DEG F OR DEG C>
*
C   *    CURVE 2 - OUTSIDE AIR TEMPERATURE (WB) IN <DEG F OR DEG C>
*
C   *    CURVE 3 - INSIDE AIR TEMPERATURE (DB) IN <DEG F OR DEG C>
*
C   *    CURVE 4 - INSIDE AIR TEMPERATURE (WB) IN <DEG F OR DEG C>
*
C   *    CURVE 5 - SOLAR INSOLATION (HORIZONTAL) IN
*
C   *        <BTU/HR-SQFT OR W/SQ-M>
*
C   *
*
C   ************************************************************************
*
C
      GO TO (1100,1200,1300,1400,1500),PLOT
C
C--- BGNPL SETS UP THE FIRST PLOT FOR DISSPLA
C
 1100 CALL BGNPL(1)
      CALL NEWPEN(ONE)
      CALL MARKER(ONE)
C
C--- COMPLX CHANGES THE CHARACTER STYLE TO COMPLEX
C
      CALL COMPLX
C
C--- BASALF ALLOWS FOR THE USE OF LOWER CASE LETTERS
C
      CALL BASALF('L/CSTD')
C
C--- MIXALF ALLOWS FOR THE USE OF UPPER CASE LETTERS
C
      CALL MIXALF('STANDARD')
C
```

```
PLOT 174 C--- HEIGHT SETS THE CHARACTER HEIGHT FOR THE LEGEND
PLOT 175 C
PLOT 176       CALL HEIGHT(0.10)
PLOT 177 C
PLOT 178 C--- THE FOLLWING 5 CALL TO LINES FILLS THE LEGEND
PLOT 179 C
PLOT 180       CALL LINES('(O)UTSIDE (DB)$',LEGND,1)
PLOT 181       CALL LINES('(O)UTSIDE (WB)$',LEGND,2)
PLOT 182       CALL LINES('(I)NSIDE (DB)$',LEGND,3)
PLOT 183       CALL LINES('(I)NSIDE (WB)$',LEGND,4)
PLOT 184       CALL LINES('(S)OLAR (I)NSOLATION (()30())$',LEGND,5)
PLOT 185 C
PLOT 186 C--- XLEGND AND YLEGND DETERMINE THE SIZE OF THE LEGEND
PLOT 187 C
PLOT 188       X=XLEGND(LEGND,5)+0.2
PLOT 189       Y=YLEGND(LEGND,5)+0.2
PLOT 190 C
PLOT 191 C--- RESET CHANGES THE CHARACTER HEIGHT BACK FOR THE TITLE
PLOT 192 C
PLOT 193       CALL RESET('HEIGHT')
PLOT 194 C
PLOT 195 C--- YINTAX AND YAXANG CHANGES THE TYPE OF YAXES
PLOT 196 C
PLOT 197       CALL YINTAX
PLOT 198       CALL YAXANG(0.)
PLOT 199 C
PLOT 200 C--- YTICKS AND XTICKS SET THE NUMBER OF TICKS FOR THE X AND Y AXES
PLOT 201 C
PLOT 202       CALL YTICKS(2)
PLOT 203       CALL XTICKS(1)
PLOT 204 C
PLOT 205 C--- TITLE SETS THE METRIC TITLE
PLOT 206 C
PLOT 207       CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
PLOT 208      >           '(T)EMPERATURE (()(D)EG (C)())',31,7.,5.)
PLOT 209 C
PLOT 210 C--- HEADIN PRINTS THE PLOT HEADING
PLOT 211 C
PLOT 212       CALL HEADIN('(H)OUSE AND (W)EATHER (D)ATA',28,1.,2)
PLOT 213       CALL HEADIN(DATE,28,1.,2)
PLOT 214 C
PLOT 215 C--- BLNK1 BLANKS OUT THE AREA FOR THE LEGEND
PLOT 216 C
PLOT 217       CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 218 C
PLOT 219 C--- XLABGR SETS THE X AND Y AXES FOR A METRIC PLOT
PLOT 220 C
PLOT 221       CALL XLABGR(DAYS,2,8,-20.,10.,50.)
PLOT 222 C
PLOT 223 C--- CURVE PLOTS THE FIRST CURVE - OUTSIDE DB
PLOT 224 C
PLOT 225  1180 CALL NEWPEN(TWO)
PLOT 226       CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 227       RETURN
PLOT 228 C
PLOT 229 C--- DOT CHANGES THE LINE PATTERN TO DOTS AND PLOTS
PLOT 230 C--- CURVE 2 - OUTSIDE WB
PLOT 231 C
```

```
PLOT 232  1200 CALL DOT
PLOT 233       CALL MARKER(TWO)
PLOT 234       GO TO 1410
PLOT 235 C
PLOT 236 C--- DASH CHANGES THE LINE PATTERN TO DASH AND PLOTS
PLOT 237 C--- CURVE 3 - INSIDE DB
PLOT 238 C
PLOT 239  1300 CALL DASH
PLOT 240       CALL MARKER(THREE)
PLOT 241       CALL NEWPEN(THREE)
PLOT 242       GO TO 1410
PLOT 243 C
PLOT 244 C--- CHNDOT CHANGES THE LINE PATTERN TO CHAINDOT AND PLOTS
PLOT 245 C--- CURVE 4 - INSIDE WB
PLOT 246 C
PLOT 247  1400 CALL CHNDOT
PLOT 248       CALL MARKER(FOUR)
PLOT 249 C
PLOT 250 C--- CURVE PLOTS CURVES 2,3, AND 4
PLOT 251 C
PLOT 252  1410 CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 253       RETURN
PLOT 254 C
PLOT 255 C--- CHNDSH CHANGES THE LINE PATTERN TO CHAINDASH AND PLOTS
PLOT 256 C--- CURVE 5 - SOLAR INSOLATION (HORIZONTAL)
PLOT 257 C
PLOT 258  1500 CALL CHNDSH
PLOT 259       CALL MARKER(FIVE)
PLOT 260       CALL NEWPEN(ONE)
PLOT 261 C
PLOT 262 C--- XPOSN AND YPOSN DETERMINE THE POSITION FOR THE SECONDARY AXES
PLOT 263 C
PLOT 264       XPOS=XPOSN(8.,-20.)
PLOT 265       YPOS=YPOSN(8.,-20.)
PLOT 266 C
PLOT 267 C--- YGRAXS DRAWS THE SECONDARY ACES TN METRIC UNITS
PLOT 268 C
PLOT 269       CALL YGRAXS(0.,1.,10.,5.,
PLOT 270      >            '(S)OLAR (I)NSOLATION (()(W/SQM)*100())',-38,
PLOT 271      >            XPOS,YPOS)
PLOT 272 C
PLOT 273 C--- CURVE PLOTS CURVE 5 - SOLAR INSOLATION (HORIZONTAL)
PLOT 274 C
PLOT 275       DO 1510 I=1,NOPTS
PLOT 276       YARRAY(I)=YARRAY(I)/100.
PLOT 277  1510 CONTINUE
PLOT 278       CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 279 C
PLOT 280 C--- HEIGHT SETS THE CHARACTER HEIGHT FOR THE LEGEND
PLOT 281 C
PLOT 282       CALL HEIGHT(0.10)
PLOT 283 C
PLOT 284 C--- LEGEND DRAWS THE LEGEND FOR PLOT 1
PLOT 285 C
PLOT 286       CALL RESET('BLNKS')
PLOT 287       Y=Y-0.1
PLOT 288       CALL LEGEND(LEGND,5,0.1,5.-Y)
PLOT 289 C
```

```
PLOT 290 C--- FRAME DRAWS THE FRAME AROUND PLOT 1
PLOT 291 C
PLOT 292       CALL FRAME
PLOT 293 C
PLOT 294 C--- ENDPL ENDS PLOT 1 FOR DISSPLA
PLOT 295 C
PLOT 296       CALL ENDPL(1)
PLOT 297 C
PLOT 298 C   ************************************************************************
*
PLOT 299 C   ****************************              **************************
*
PLOT 300 C   **************************** END OF PLOT1 **************************
*
PLOT 301 C   ****************************              **************************
*
PLOT 302 C   ************************************************************************
*
PLOT 303 C
PLOT 304       RETURN
PLOT 305       ENTRY PLTNO2 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 306 C
PLOT 307 C   ************************************************************************
*
PLOT 308 C   ******************************        ******************************
*
PLOT 309 C   ****************************** PLTNO2 ******************************
*
PLOT 310 C   ******************************        ******************************
*
PLOT 311 C   ************************************************************************
*
PLOT 312 C   *
*
PLOT 313 C   *    ENTRY - PLTNO2 PRODUCES THE SECOND PLOT OF THE ACES DATA.
*
PLOT 314 C   *    THIS PLOT DRAWS THE FOLLOWING 5 CURVES.
*
PLOT 315 C   *
*
PLOT 316 C   *    CURVE 1 - HOUSE HEATING LOAD IN <BTU/HR OR KW>
*
PLOT 317 C   *    CURVE 2 - HOUSE COOLING LOAD IN <BTU/HR OR KW>
*
PLOT 318 C   *    CURVE 3 - HOT WATER HEATING LOAD IN <BTU/HR OR KW>
*
PLOT 319 C   *    CURVE 4 - ECONOMY COOLING LOAD IN <BTU/HR OR KW>
*
PLOT 320 C   *    CURVE 5 - TOTAL HOUSE LOAD IN <BTU/HR OR KW>
*
PLOT 321 C   *
*
PLOT 322 C   ************************************************************************
*
PLOT 323 C
PLOT 324       GO TO (2100,2200,2300,2400,2500),PLOT
PLOT 325  2100 CALL BGNPL(2)
PLOT 326       CALL MARKER(ONE)
```

```
PLOT 327        CALL COMPLX
PLOT 328        CALL BASALF('L/CSTD')
PLOT 329        CALL MIXALF('STANDARD')
PLOT 330        CALL HEIGHT(0.10)
PLOT 331        CALL LINES('(H)EATING (L)OAD$',LEGND,1)
PLOT 332        CALL LINES('(C)OOLING (L)OAD$',LEGND,2)
PLOT 333        CALL LINES('(H)OT (W)ATER (L)OAD$',LEGND,3)
PLOT 334        CALL LINES('(E)CONOMY (C)OOLING (L)OAD$',LEGND,4)
PLOT 335        CALL LINES('(T)OTAL (L)OAD$',LEGND,5)
PLOT 336        X=XLEGND(LEGND,5)+0.2
PLOT 337        Y=YLEGND(LEGND,5)+0.2
PLOT 338        CALL RESET('HEIGHT')
PLOT 339        CALL YTICKS(2)
PLOT 340        CALL XTICKS(1)
PLOT 341        CALL YINTAX
PLOT 342        CALL YAXANG(0.)
PLOT 343        CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
PLOT 344       >           '(B)UILDING (L)OADS (()(KW)())',29,7.,5.)
PLOT 345        CALL XLABGR(DAYS,2,8,0.,1.,16.)
PLOT 346 C
PLOT 347 C--- THIS LOOP CONVERTS FROM WATTS TO KW
PLOT 348 C
PLOT 349        DO 2110 I=1,NOPTS
PLOT 350        YARRAY(I)=YARRAY(I)/1000.
PLOT 351  2110 CONTINUE
PLOT 352        CALL HEADIN('(ACES L)OADS',12,1.,2)
PLOT 353        CALL HEADIN(DATE,28,1.,2)
PLOT 354        CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 355        CALL NEWPEN(TWO)
PLOT 356        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 357        RETURN
PLOT 358  2200 CALL DASH
PLOT 359        CALL MARKER(TWO)
PLOT 360        GO TO 2510
PLOT 361  2300 CALL DOT
PLOT 362        CALL MARKER(THREE)
PLOT 363        CALL NEWPEN(THREE)
PLOT 364        GO TO 2510
PLOT 365  2400 CALL CHNDSH
PLOT 366        CALL MARKER(FOUR)
PLOT 367        GO TO 2510
PLOT 368  2500 CALL CHNDOT
PLOT 369        CALL MARKER(FIVE)
PLOT 370        CALL NEWPEN(ONE)
PLOT 371  2510 DO 2520 I=1,NOPTS
PLOT 372        YARRAY(I)=YARRAY(I)/1000.
PLOT 373  2520 CONTINUE
PLOT 374        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 375        IF (PLOT .EQ. 5) GO TO 2540
PLOT 376        RETURN
PLOT 377  2540 CALL HEIGHT(0.10)
PLOT 378        CALL RESET('BLNKS')
PLOT 379        Y=Y-0.1
PLOT 380        CALL LEGEND(LEGND,5,0.1,5.-Y)
PLOT 381        CALL FRAME
PLOT 382        CALL ENDPL(2)
PLOT 383 C
```

```
PLOT 384 C   ************************************************************************
*
PLOT 385 C   ****************************           ****************************
*
PLOT 386 C   **************************** END OF PLOT2 ****************************
*
PLOT 387 C   ****************************           ****************************
*
PLOT 388 C   ************************************************************************
*
PLOT 389 C
PLOT 390       RETURN
PLOT 391       ENTRY PLTNO3 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 392 C
PLOT 393 C   ************************************************************************
*
PLOT 394 C   ********************************       ********************************
*
PLOT 395 C   ******************************** PLTNO3 ********************************
*
PLOT 396 C   ********************************       ********************************
*
PLOT 397 C   ************************************************************************
*
PLOT 398 C   *
*
PLOT 399 C   *    ENTRY - PLTNO3 PRODUCES THE THIRD PLOT OF THE ACES DATA.
*
PLOT 400 C   *    THIS PLOT DRAWS THE FOLLOWING 3 CURVES.
*
PLOT 401 C   *
*
PLOT 402 C   *    CURVE 1 - WIND SPEED IN <MPH OR KPH>
*
PLOT 403 C   *    CURVE 2 - SOLAR INSOLATION (HORIZONTAL) IN
*
PLOT 404 C   *              <BTU/HR-SQFT OR W/SQM*100>
*
PLOT 405 C   *    CURVE 3 - PANEL LOADS IN <BTU/HR*1000 OR KW>
*
PLOT 406 C   *
*
PLOT 407 C   ************************************************************************
*
PLOT 408 C
PLOT 409       GO TO (3100,3300,3400),PLOT
PLOT 410  3100 CALL BGNPL(3)
PLOT 411       CALL MARKER(ONE)
PLOT 412       CALL COMPLX
PLOT 413       CALL BASALF('L/CSTD')
PLOT 414       CALL MIXALF('STANDARD')
PLOT 415       CALL HEIGHT(0.10)
PLOT 416       CALL LINES('(W)IND (S)PEED$',LEGND,1)
PLOT 417       CALL LINES('(S)OLAR (I)NSOLATION (()(H)())$',LEGND,2)
PLOT 418       CALL LINES('(P)ANEL (L)OADS$',LEGND,3)
PLOT 419       X=XLEGND(LEGND,3)+0.2
PLOT 420       Y=YLEGND(LEGND,3)+0.2
PLOT 421       CALL RESET('HEIGHT')
```

```
PLOT 422        CALL YTICKS(2)
PLOT 423        CALL XTICKS(1)
PLOT 424        CALL YINTAX
PLOT 425        CALL YAXANG(0.)
PLOT 426        CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
PLOT 427       >           '(W)IND (S)PEED (()(KPH)())',
PLOT 428       >           26,7.,5.)
PLOT 429        CALL XLABGR(DAYS,2,8,0.,20.,100.)
PLOT 430        CALL HEADIN('(S)OLAR (P)ANEL (A)NALYSIS',26,1.,2)
PLOT 431        CALL HEADIN(DATE,28,1.,2)
PLOT 432        CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 433        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 434        RETURN
PLOT 435   3300 CALL DOT
PLOT 436        CALL MARKER(TWO)
PLOT 437        XPOS=XPOSN(8.,0.)
PLOT 438        YPOS=YPOSN(8.,0.)
PLOT 439        CALL YGRAXS(0.,1.,16.,5.,'(W/SQM)*100 OR (KW)',-19,XPOS,YPOS)
PLOT 440 C
PLOT 441 C--- THIS LOOP CONVERTS FROM W/SQM TO W/SQM*100
PLOT 442 C
PLOT 443        DO 3310 I=1,NOPTS
PLOT 444        YARRAY(I)=YARRAY(I)/100.
PLOT 445   3310 CONTINUE
PLOT 446        CALL NEWPEN(THREE)
PLOT 447        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 448        RETURN
PLOT 449   3400 CALL CHNDSH
PLOT 450        CALL MARKER(THREE)
PLOT 451        CALL NEWPEN(TWO)
PLOT 452 C
PLOT 453 C--- THIS LOOP CONVERTS FROM WATTS TO KW
PLOT 454 C
PLOT 455        DO 3410 I=1,NOPTS
PLOT 456        YARRAY(I)=YARRAY(I)/1000.
PLOT 457   3410 CONTINUE
PLOT 458        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 459        CALL HEIGHT(0.10)
PLOT 460        CALL RESET('BLNKS')
PLOT 461        Y=Y-0.1
PLOT 462        CALL NEWPEN(ONE)
PLOT 463        CALL LEGEND(LEGND,5,0.1,5.-Y)
PLOT 464        CALL FRAME
PLOT 465        CALL ENDPL(3)
PLOT 466 C
PLOT 467 C   ************************************************************************
*
PLOT 468 C   ****************************              ************************
*
PLOT 469 C   **************************** END OF PLOT3 ************************
*
PLOT 470 C   ****************************              ************************
*
PLOT 471 C   ************************************************************************
*
```

```
PLOT 472 C
PLOT 473       RETURN
PLOT 474       ENTRY PLTNO4 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 475 C
PLOT 476 C   ************************************************************************
*
PLOT 477 C   ******************************        ******************************
*
PLOT 478 C   ****************************** PLTNO4 ******************************
*
PLOT 479 C   ******************************        ******************************
*
PLOT 480 C   ************************************************************************
*
PLOT 481 C   *
*
PLOT 48  C   *    ENTRY - PLTNO4 PRODUCES THE FOURTH PLOT OF THE ACES DATA.
*
PLOT 483 C   *    THIS PLOT DRAWS THE FOLLOWING 5 CURVES.
*
PLOT 485 C   *    CURVE 1 - PUMP POWER CONSUMPTION IN KWH
*
PLOT 486 C   *    CURVE 2 - COMPRESSOR POWER CONSUMPTION IN KWH
*
*PLOT 48 C   *    CURVE 3 - FANS POWER CONSUMPTION IN KWH
PLOT 488 C   *    CURVE 4 - ACES POWER CONSUMPTION IN KWH
*
PLOT 489 C   *    CURVE 5 - TOTAL POWER CONSUMPTION IN KWH
*
PLOT 490 C   *
*
PLOT 491 C   ************************************************************************
*
PLOT 492 C
PLOT 493       DO 4000 I=1,NOPTS
PLOT 494       YARRAY(I)=YARRAY(I)/1000.
PLOT 495  4000 CONTINUE
PLOT 496       GO TO (4100,4200,4300,4400,4500),PLOT
PLOT 497  4100 CALL BGNPL(4)
PLOT 498       CALL MARKER(ONE)
PLOT 499       CALL COMPLX
PLOT 500       CALL BASALF('L/CSTD')
PLOT 501       CALL MIXALF('STANDARD')
PLOT 502       CALL HEIGHT(0.10)
PLOT 503       CALL LINES('(P)UMPS$',LEGND,1)
PLOT 504       CALL LINES('(C)OMPRESSER$',LEGND,2)
PLOT 50       CALL LINES('(F)ANS$',LEGND,3)
PLOT 506       CALL LINES('(ACES)$',LEGND,4)
PLOT 507       CALL LINES('(H)OUSE$',LEGND,5)
PLOT 508       X=XLEGND(LEGND,5)+0.2
PLOT 509       Y=YLEGND(LEGND,5)+0.2
PLOT 510       CALL RESET('HEIGHT')
PLOT 511       CALL YTICKS(2)
PLOT 512       CALL XTICKS(1)
PLOT 513       CALL YINTAX
PLOT 514       CALL YAXANG(0.)
PLOT 515       CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
```

```
PLOT 516      >           '(P)OWER (C)ONSUMPTION (()(KW)())',32,7.,5.)
PLOT 517       CALL XLABGR(DAYS,2,8,0.,1.,16.)
PLOT 518       CALL HEADIN('(ACES E)LECTRICAL (A)NALYSIS',28,1.,2)
PLOT 519       CALL HEADIN(DATE,28,1.,2)
PLOT 520       CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 521       CALL NEWPEN(TWO)
PLOT 522       CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 523       RETURN
PLOT 524  4200 CALL DASH
PLOT 525       CALL MARKER(TWO)
PLOT 526       GO TO 4510
PLOT 527  4300 CALL DOT
PLOT 528       CALL MARKER(THREE)
PLOT 529       CALL NEWPEN(THREE)
PLOT 530       GO TO 4510
PLOT 531  4400 CALL CHNDSH
PLOT 532       CALL MARKER(FOUR)
PLOT 533       GO TO 4510
PLOT 534  4500 CALL CHNDOT
PLOT 535       CALL NEWPEN(ONE)
PLOT 536       CALL MARKER(FIVE)
PLOT 537  4510 CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 538       IF (PLOT .EQ. 5) GO TO 4520
PLOT 539       RETURN
PLOT 540  4520 CALL HEIGHT(0.10)
PLOT 541       CALL RESET('BLNKS')
PLOT 542       Y=Y-0.1
PLOT 543       CALL LEGEND(LEGND,5,0.1,5.-Y)
PLOT 544       CALL FRAME
PLOT 545       CALL ENDPL(4)
PLOT 546 C
PLOT 547 C   ************************************************************************
*
PLOT 548 C   ****************************              ************************
*
PLOT 549 C   **************************** END OF PLOT4 ************************
*
PLOT 550 C   ****************************              ************************
*
PLOT 551 C   ************************************************************************
*
PLOT 552 C
PLOT 553       RETURN
PLOT 554       ENTRY PLTNO5 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 555 C
PLOT 556 C   ************************************************************************
*
PLOT 557 C   ******************************        ******************************
*
PLOT 558 C   ****************************** PLTNO5 ******************************
*
PLOT 559 C   ******************************        ******************************
*
PLOT 560 C   ************************************************************************
*
PLOT 561 C   *
*
```

```
PLOT 562 C   *     ENTRY - PLTNO5 PRODUCES THE FIFTH PLOT OF THE ACES DATA
*
PLOT 563 C   *     THIS PLOT DRAWS THE FOLLOWING 3 CURVES
*
PLOT 564 C   *
*
PLOT 565 C   *     CURVE 1 - COMPRESSOR COP
*
PLOT 566 C   *     CURVE 2 - SYSTEM COP
*
PLOT 567 C   *     CURVE 3 - TOTAL LOAD IN <BTU/HR OR KW>
*
PLOT 568 C   *
*
PLOT 569 C   ************************************************************************
*
PLOT 570 C
PLOT 571        GO TO (5100,5200,5300),PLOT
PLOT 572  5100 CALL BGNPL(5)
PLOT 573        CALL MARKER(ONE)
PLOT 574        CALL COMPLX
PLOT 575        CALL BASALF('L/CSTD')
PLOT 576        CALL MIXALF('STANDARD')
PLOT 577        CALL HEIGHT(0.10)
PLOT 578        CALL LINES('(C)OMPRESSER (O)NLY (COP)$',LEGND,1)
PLOT 579        CALL LINES('(S)YSTEM (COP)$',LEGND,2)
PLOT 580        CALL LINES('(S)YSTEM (L)OAD$',LEGND,3)
PLOT 581        X=XLEGND(LEGND,3)+0.2
PLOT 582        Y=YLEGND(LEGND,3)+0.2
PLOT 583        CALL RESET('HEIGHT')
PLOT 584        CALL XTICKS(1)
PLOT 585        CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
PLOT 586       >           '(COP)',0,7.,5.)
PLOT 587        CALL XLABGR(DAYS,2,8,0.,1.,10.)
PLOT 588        CALL HEADIN('(ACES P)ERFORMANCE (A)NALYSIS',29,1.,2)
PLOT 589        CALL HEADIN(DATE,28,1.,2)
PLOT 590        CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 591        CALL ANGLE(90.)
PLOT 592        CALL MESSAG('(COP)',5,-.5,2.3)
PLOT 593        CALL ANGLE(0.)
PLOT 594        CALL VECTOR(-0.05,0.0,0.0,0.0,0)
PLOT 595        CALL VECTOR(-0.05,0.5,0.0,0.5,0)
PLOT 596        CALL VECTOR(-0.05,1.0,0.0,1.0,0)
PLOT 597        CALL VECTOR(-0.05,1.5,0.0,1.5,0)
PLOT 598        CALL VECTOR(-0.05,2.0,0.0,2.0,0)
PLOT 599        CALL VECTOR(-0.05,2.5,7.0,2.5,0)
PLOT 600        CALL VECTOR(-0.05,3.0,0.0,3.0,0)
PLOT 601        CALL VECTOR(-0.05,3.5,0.0,3.5,0)
PLOT 602        CALL VECTOR(-0.05,4.0,0.0,4.0,0)
PLOT 603        CALL VECTOR(-0.05,4.5,0.0,4.5,0)
PLOT 604        CALL VECTOR(-0.05,5.0,0.0,5.0,0)
PLOT 605        CALL HEIGHT(0.10)
PLOT 606        CALL RLINT(0,0.8,0.0-.10)
PLOT 607        CALL RLINT(1,0.8,1.0-.10)
PLOT 608        CALL RLINT(2,0.8,2.0-.10)
PLOT 609        CALL RLINT(3,0.8,3.0-.10)
PLOT 610        CALL RLINT(4,0.8,4.0-.10)
```

```
PLOT 611        CALL RLINT(5,0.8,5.0-.10)
PLOT 612        CALL RLINT(25,0.7,6.0-.10)
PLOT 613        CALL RLINT(45,0.7,7.0-.10)
PLOT 614        CALL RLINT(65,0.7,8.0-.10)
PLOT 615        CALL RLINT(85,0.7,9.0-.10)
PLOT 616        CALL RLINT(105,0.6,10.0-.10)
PLOT 617        CALL RESET('HEIGHT')
PLOT 618        CALL NEWPEN(TWO)
PLOT 619   5110 DO 5130 I=1,NOPTS
PLOT 620        IF (YARRAY(I)-5.) 5130,5130,5120
PLOT 621   5120 YARRAY(I)=(YARRAY(I)-5.)/20.+5.
PLOT 622   5130 CONTINUE
PLOT 623        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 624        RETURN
PLOT 625   5200 CALL DASH
PLOT 626        CALL MARKER(TWO)
PLOT 627        CALL NEWPEN(THREE)
PLOT 628        GO TO 5110
PLOT 629   5300 CALL DOT
PLOT 630        CALL MARKER(THREE)
PLOT 631        CALL YINTAX
PLOT 632        CALL YAXANG(0.)
PLOT 633        CALL NEWPEN(ONE)
PLOT 634        XPOS=XPOSN(8.,0.)
PLOT 635        YPOS=YPOSN(8.,0.)
PLOT 636        CALL YGRAXS(0.,1.,16.,5.,'(S)YSTEM (L)OADS (()(KW)())',-27,
PLOT 637       >             XPOS,YPOS)
PLOT 638 C
PLOT 639 C--- THIS LOOP CONVERTS FROM WATTS TO KW
PLOT 640 C
PLOT 641        DO 5310 I=1,NOPTS
PLOT 642        YARRAY(I)=YARRAY(I)/1000.
PLOT 643   5310 CONTINUE
PLOT 644        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 645        CALL HEIGHT(0.10)
PLOT 646        CALL RESET('BLNKS')
PLOT 647        Y=Y-0.1
PLOT 648        CALL LEGEND(LEGND,3,0.1,5.-Y)
PLOT 649        CALL FRAME
PLOT 650        CALL ENDPL(5)
PLOT 651 C
PLOT 652 C    ************************************************************************
*
PLOT 653 C    ****************************              ***************************
*
PLOT 654 C    **************************** END OF PLOT5 ***************************
*
PLOT 655 C    ****************************              ***************************
*
PLOT 656 C    ************************************************************************
*
PLOT 657 C
PLOT 658        RETURN
PLOT 659        ENTRY PLTNO6 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 660 C
```

```
PLOT 661 C   ************************************************************************
*
PLOT 662 C   ****************************        ****************************
*
PLOT 663 C   **************************** PLTNO6 ****************************
*
PLOT 664 C   ****************************        ****************************
*
PLOT 665 C   ************************************************************************
*
PLOT 666 C   *
*
PLOT 667 C   *     ENTRY - PLTNO6 PRODUCES THE SIXTH PLOT OF THE ACES DATA.
*
PLOT 668 C   *     THIS PLOT DRAWS THE FOLLOWING 5 CURVES,
*
PLOT 669 C   *
*
PLOT 670 C   *     CURVE 1 - OUTSIDE AIR TEMPERATURE (DB) IN <DEG F OR DEG C>
*
PLOT 671 C   *     CURVE 2 - OUTSIDE AIR TEMPERATURE (WB) IN <DEG F OR DEG C>
*
PLOT 672 C   *     CURVE 3 - INSIDE AIR TEMPERATURE (DB) IN <DEG F OR DEG C>
*
PLOT 673 C   *     CURVE 4 - INSIDE AIR TEMPERATURE (WB) IN <DEG F OR DEG C>
*
PLOT 674 C   *     CURVE 5 - HOUSE LOADS IN <BTU/HR OR KW>
*
PLOT 675 C   *
*
PLOT 676 C   ************************************************************************
*
PLOT 677 C
PLOT 678        GO TO (6100,6200,6300,6400,6500),PLOT
PLOT 679   6100 CALL BGNPL(6)
PLOT 680        CALL MARKER(ONE)
PLOT 681        CALL COMPLX
PLOT 682        CALL BASALF('L/CSTD')
PLOT 683        CALL MIXALF('STANDARD')
PLOT 684        CALL HEIGHT(0.10)
PLOT 685        CALL LINES('(O)UTSIDE (DB)$',LEGND,1)
PLOT 686        CALL LINES('(O)UTSIDE (WB)$',LEGND,2)
PLOT 687        CALL LINES('(I)NSIDE (DB)$',LEGND,3)
PLOT 688        CALL LINES('(I)NSIDE (WB)$',LEGND,4)
PLOT 689        CALL LINES('(H)OUSE (L)OADS$',LEGND,5)
PLOT 690        X=XLEGND(LEGND,5)+0.2
PLOT 691        Y=YLEGND(LEGND,5)+0.2
PLOT 692        CALL RESET('HEIGHT')
PLOT 693        CALL YINTAX
PLOT 694        CALL YAXANG(0.)
PLOT 695        CALL YTICKS(2)
PLOT 696        CALL XTICKS(1)
PLOT 697        CALL TITLE(' ',-1,'(D)AYS OF THE (W)EEK',20,
PLOT 698       >           '(T)EMPERATURE (()(D)EG (C)())',31,7.,5.)
PLOT 699        CALL HEADIN('(C)ONTROL (H)OUSE AND (W)EATHER (D)ATA',38,1.,2)
PLOT 700        CALL HEADIN(DATE,28,1.,2)
PLOT 701        CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 702        CALL XLABGR(DAYS,2,8,-20.,10..50.)
```

```
PLOT 703        CALL NEWPEN(TWO)
PLOT 704        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 705        RETURN
PLOT 706   6200 CALL DOT
PLOT 707        CALL MARKER(TWO)
PLOT 708        GO TO 6310
PLOT 709   6300 CALL DASH
PLOT 710        CALL MARKER(THREE)
PLOT 711        CALL NEWPEN(THREE)
PLOT 712   6310 CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 713        RETURN
PLOT 714   6400 CALL CHNDOT
PLOT 715        CALL MARKER(FOUR)
PLOT 716        GO TO 6310
PLOT 717   6500 CALL CHNDSH
PLOT 718        CALL MARKER(FIVE)
PLOT 719        CALL NEWPEN(ONE)
PLOT 720        XPOS=XPOSN(8.,-20.)
PLOT 721        YPOS=YPOSN(8.,-20.)
PLOT 722        CALL YGRAXS(0.,1.,16.,5.,
 PLOT       >            '(P)OWER (C)ONSUMPTION (()(KW)())',-32,
PLOT 724      >             XPOS,YPOS)
PLOT 725 C
PLOT 726 C--- THIS LOOP CONVERTS FROM WATTS TO KW
PLOT 727 C
PLOT 728        DO 6410 I=1,NOPTS
PLOT 729        YARRAY(I)=YARRAY(I)/1000.
PLOT 730   6410 CONTINUE
PLOT 731   6430 CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 732        CALL HEIGHT(0.10)
PLOT 733        CALL RESET('BLNKS')
PLOT 734        Y=Y-0.1
PLOT 735        CALL LEGEND(LEGND,5,0.1,5.-Y)
PLOT 736        CALL FRAME
PLOT 737        CALL ENDPL(6)
PLOT 738 C
PLOT 739 C   ************************************************************************
*
PLOT 740 C   ****************************              **************************
*
PLOT 741 C   **************************** END OF PLOT6 **************************
*
PLOT 742 C   ****************************              **************************
*
PLOT 743 C   ************************************************************************
*
PLOT 744 C
PLOT 745        RETURN
PLOT 746        ENTRY PLTNO7 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 747 C
PLOT 748 C   ************************************************************************
*
PLOT 749 C   ********************************        ******************************
*
PLOT 750 C   ******************************** PLTNO7 ******************************
*
PLOT 751 C   ********************************        ******************************
*
```

```
PLOT 752 C   ***********************************************************************
*
PLOT 753 C   *
*
PLOT 754 C   *    ENTRY - PLTNO7 PRODUCES THE SEVENTH PLOT OF THE ACES DATA.
*
PLOT 755 C   *    THIS PLOT DRAWS THE FOLLOWING TWO CURVES.
*
PLOT 756 C   *
*
PLOT 757 C   *    CURVE 1 - ACES POWER CONSUMPTION IN KWH
*
PLOT 758 C   *    CURVE 2 - CONTROL HOUSE POWER CONSUMPTION IN KWH
*
PLOT 759 C   *
*
PLOT 760 C   ***********************************************************************
*
PLOT 761 C
PLOT 762        DO 7000 I=1,NOPTS
PLOT 763        YARRAY(I)=YARRAY(I)/1000.
PLOT 764  7000 CONTINUE
PLOT 765        GO TO (7100,7200),PLOT
PLOT 766  7100 CALL BGNPL(7)
PLOT 767        CALL MARKER(ONE)
PLOT 768        CALL COMPLX
PLOT 769        CALL BASALF('L/CSTD')
PLOT 770        CALL MIXALF('STANDARD')
PLOT 771        CALL HEIGHT(0.10)
PLOT 772        CALL LINES('(ACES)$',LEGND,1)
PLOT 773        CALL LINES('(C)ONTROL (H)OUSE$',LEGND,2)
PLOT 774        X=XLEGND(LEGND,2)+0.2
PLOT 775        Y=YLEGND(LEGND,2)+0.2
PLOT 776        CALL RESET('HEIGHT')
PLOT 777        CALL YTICKS(2)
PLOT 778        CALL XTICKS(1)
PLOT 779        CALL YINTAX
PLOT 780        CALL YAXANG(0.)
PLOT 781        CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
PLOT 782       >           '(P)OWER (C)ONSUMPTION (()(KW)())',32,7.,5.)
PLOT 783        CALL XLABGR(DAYS,2,8,0.,1.,16.)
PLOT 784        CALL HEADIN('(C)OMPARISON OF (ACES) AND (C)ONTROL (H)OUSE',
PLOT 785       >            44,1.,2)
PLOT 786        CALL HEADIN(DATE,28,1.,2)
PLOT 787        CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 788        CALL NEWPEN(TWO)
PLOT 789        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 790        RETURN
PLOT 791  7200 CALL DASH
PLOT 792        CALL MARKER(TWO)
PLOT 793        CALL NEWPEN(THREE)
PLOT 794        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 795        CALL NEWPEN(ONE)
PLOT 796        CALL HEIGHT(0.10)
PLOT 797        CALL RESET('BLNKS')
PLOT 798        Y=Y-0.1
PLOT 799        CALL LEGEND(LEGND,2,0.1,5.-Y)
PLOT 800        CALL FRAME
```

```
PLOT 801        CALL ENDPL(7)
PLOT 802        RETURN
PLOT 803 C
PLOT 804 C   ************************************************************************
*
PLOT 805 C   ****************************              ****************************
*
PLOT 806 C   **************************** END OF PLOT7 ****************************
*
PLOT 807 C   ****************************              ****************************
*
PLOT 808 C   ************************************************************************
*
PLOT 809 C
PLOT 810        ENTRY PLTNO8 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
PLOT 811 C
PLOT 812 C   ************************************************************************
*
PLOT 813 C   ******************************        ******************************
*
PLOT 814 C   ****************************** PLTNO8 ******************************
*
PLOT 815 C   ******************************        ******************************
*
PLOT 816 C   ************************************************************************
*
PLOT 817 C   *
*
PLOT 818 C   *    ENTRY - PLTNO8 PRODUCES THE EIGHTH PLOT OF THE ACES DATA.
*
PLOT 819 C   *    THIS PLOT DRAWS THE FOLLOWING THREE CURVES.
*
PLOT 820 C   *
*
PLOT 821 C   *    CURVE 1 - ACES POWER CONSUMPTION IN KWH
*
PLOT 822 C   *    CURVE 2 - CONTROL HOUSE POWER CONSUMPTION IN KWH
*
PLOT 823 C   *    CURVE 3 - SOLAR HOUSE POWER CONSUMPTION IN KWH
*
PLOT 824 C   *
*
PLOT 825 C   ************************************************************************
*
PLOT 826 C
PLOT 827        DO 8000 I=1,NOPTS
PLOT 828        YARRAY(I)=YARRAY(I)/1000.
PLOT 829  8000 CONTINUE
PLOT 830        GO TO (8100,8200,8300),PLOT
PLOT 831  8100 CALL BGNPL(8)
PLOT 832        CALL MARKER(ONE)
PLOT 833        CALL COMPLX
PLOT 834        CALL BASALF('L/CSTD')
PLOT 835        CALL MIXALF('STANDARD')
PLOT 836        CALL HEIGHT(0.10)
PLOT 837        CALL LINES('(ACES)$',LEGND,1)
PLOT 838        CALL LINES('(C)ONTROL (H)OUSE$',LEGND,2)
PLOT 839        CALL LINES('(S)OLAR (H)OUSE$',LEGND,3)
```

```
PLOT 840        X=XLEGND(LEGND,3)+0.2
PLOT 841        Y=YLEGND(LEGND,3)+0.2
PLOT 842        CALL RESET('HEIGHT')
PLOT 843        CALL YTICKS(2)
PLOT 844        CALL XTICKS(1)
PLOT 845        CALL YINTAX
PLOT 846        CALL YAXANG(0.)
PLOT 847        CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
PLOT 848       >           '(P)OWER (C)ONSUMPTION (()(KW)())',32,7.,5.)
PLOT 849        CALL XLABGR(DAYS,2,8,0.,1.,16.)
PLOT 850        CALL HEADIN('(C)OMP OF (ACES), (S)OLAR, AND (C)ONTROL (H)OUSE',
PLOT 851       >            48,1.,2)
PLOT 852        CALL HEADIN(DATE,28,1.,2)
PLOT 853        CALL BLNK1(0.,X,5.-Y,5.,2)
PLOT 854        CALL NEWPEN(TWO)
PLOT 855        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 856        RETURN
PLOT 857   8200 CALL DASH
PLOT 858        CALL MARKER(TWO)
PLOT 859        CALL NEWPEN(THREE)
PLOT 860        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 861        CALL NEWPEN(ONE)
PLOT 862        RETURN
PLOT 863   8300 CALL DOT
PLOT 864        CALL MARKER(THREE)
PLOT 865        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
PLOT 866        CALL HEIGHT(0.10)
PLOT 867        CALL RESET('BLNKS')
PLOT 868        Y=Y-0.1
PLOT 869        CALL LEGEND(LEGND,3,0.1,5.-Y)
PLOT 870        CALL FRAME
PLOT 871        CALL ENDPL(8)
PLOT 872        RETURN
PLOT 873 C
PLOT 874 C   ************************************************************************
*
PLOT 875 C   ****************************              ************************
*
PLOT 876 C   **************************** END OF PLOT8 ************************
*
PLOT 877 C   ****************************              ************************
*
PLOT 878 C   ************************************************************************
*
PLOT 879 C
  0744 C
  0474        ENTRY PLTNO9 (XARRAY,YARRAY,NOPTS,DATE,PLOT)
  0475 C
  0476 C   ************************************************************************
  0477 C   ******************************        ******************************
  0478 C   ****************************** PLTNO9 ******************************
  0479 C   ******************************        ******************************
  0480 C   ************************************************************************
  0481 C   *                                                                      *
       C   *    ENTRY - PLTNO9 PRODUCES THE NINTH PLOT.     *          0000000
  0483 C   *    THIS PLOT DRAWS THE FOLLOWING 5 CURVES.                          *
    04 C   *    CURVE 1 - OUTDOOR UNIT CONSUMPTION IN KWH
4860000  C   *      CURVE 2 - INDOOR UNIT CONSUMPTION IN KWH                    *PLOT
```

```
0000P C   *     CURVE 3 - HOT WATER HEATER CONSUMPTION IN KWH
3630488  C   *     CURVE 4 - EQUIPMENT CONSUMPTION IN KWH                    *
P
  0489 C   *     CURVE 5 - TOTAL POWER CONSUMPTION IN KWH                    *
  0490 C   *                                                                 *
  0491 C   ***********************************************************************
  0492 C
  0493        DO 9000 I=1,NOPTS
  0494        YARRAY(I)=YARRAY(I)/1000.
  0495  9000 CONTINUE
  0496        GO TO (9100,9200,9300,9400,9500),PLOT
  0497  9100 CALL BGNPL(9)
  0498        CALL MARKER(ONE)
  0499        CALL COMPLX
  0500        CALL BASALF('L/CSTD')
  0501        CALL MIXALF('STANDARD')
  0502        CALL HEIGHT(0.10)
00000         CALL LINES('(O)UTDOOR (U)NIT$',LEGND,1)
0             CALL LINES('(I)NDOOR (U)NIT$',LEGND,2)
00000         CALL LINES('(H)OT (W)ATER (H)EATER$',LEGND,3)
000730        CALL LINES('(E)QUIPMENT$',LEGND,4)
  0507        CALL LINES('(H)OUSE$',LEGND,5)
  0508        X=XLEGND(LEGND,5)+0.2
  0509        Y=YLEGND(LEGND,5)+0.2
  0510        CALL RESET('HEIGHT')
  0511        CALL YTICKS(2)
  0512        CALL XTICKS(1)
  0513        CALL YINTAX
  0514        CALL YAXANG(0.)
  0515        CALL TITLE(' ',-1,'(T)IME (()(D)AYS())',19,
  0516       >           '(P)OWER (C)ONSUMPTION (()(KW)())',32,7.,5.)
  0517        CALL XLABGR(DAYS,2,8,0.,1.,16.)
00000         CALL HEADIN('(C)ONTROL (H)OUSE (E)LECTRICAL (A)NALYSIS',42,1.,2)
  0519        CALL HEADIN(DATE,28,1.,2)
  0520        CALL BLNK1(0.,X,5.-Y,5.,2)
  0521        CALL NEWPEN(TWO)
  0522        CALL CURVE(XARRAY,YARRAY,NOPTS,6)
  0523        RETURN
  0524  9200 CALL DASH
  0525        CALL MARKER(TWO)
  0526        GO TO 9510
  0527  9300 CALL DOT
  0528        CALL MARKER(THREE)
  0529        CALL NEWPEN(THREE)
              GO TO 9510
  0531  9400 CALL CHNDSH
  0532        CALL MARKER(FOUR)
  0533        GO TO 9510
  0534  9500 CALL CHNDOT
  0535        CALL NEWPEN(ONE)
  0536        CALL MARKER(FIVE)
  0537  9510 CALL CURVE(XARRAY,YARRAY,NOPTS,6)
  0538        IF (PLOT .EQ. 5) GO TO 9520
  0539        RETURN
  0540  9520 CALL HEIGHT(0.10)
  0541        CALL RESET('BLNKS')
  0542        Y=Y-0.1
  0543        CALL LEGEND(LEGND,5,0.1,5.-Y)
```

```
   0544        CALL FRAME
   0545        CALL ENDPL(9)
   0546 C
   0547 C    ************************************************************************
   0548 C    ****************************                ****************************
   0549 C    **************************** END OF PLOT9 ****************************
   0550 C    ****************************                ****************************
   0551 C    ************************************************************************
   0552 C
   0553        RETURN
 PLOT 880        ENTRY ENDPLT
 PLOT 882        CALL DONEPL
 PLOT 883        RETURN
 PLOT 884 C
 PLOT 885 C    ************************************************************************
 *
 PLOT 886 C    ****************************                ****************************
 *
 PLOT 887 C    **************************** END OF PLTPRG ****************************
 *
 PLOT 888 C    ****************************                ****************************
 *
 PLOT 889 C    ************************************************************************
 *
 PLOT 890 C
 PLOT 891        END
 PLOT 892        SUBROUTINE FIXUP
 PLOT 893        DIMENSION XARRAY(168),YARRAY(168),X(168),Y(168)
 PLOT 894        X(1)=1.
 PLOT 895        DELTAX=1./24.
 PLOT 896        DO 10 I=2,168
 PLOT 897        X(I)=X(I-1)+DELTAX
 PLOT 898     10 CONTINUE
 PLOT 899        RETURN
 PLOT 900        ENTRY FIXUP1 (XARRAY,YARRAY,NOPTS)
 PLOT 901        READ(10,20) Y
 PLOT 902     20 FORMAT(8F10.0)
 PLOT 903        NOPTS=0
 PLOT 904        DO 30 I=1,168
 PLOT 905        IF (Y(I) .GE. 99999.) GO TO 30
 PLOT 906        NOPTS=NOPTS+1
 PLOT 907        YARRAY(NOPTS)=Y(I)
 PLOT 908        XARRAY(NOPTS)=X(I)
 PLOT 909     30 CONTINUE
 PLOT 910        RETURN
 PLOT 911        ENTRY FIXUP0 (XARRAY,YARRAY,NOPTS)
 PLOT 912        NOPTS=168
 PLOT 913        READ(10,20) Y
 PLOT 914        DO 40 I=1,168
 PLOT 915        XARRAY(I)=X(I)
 PLOT 916        YARRAY(I)=Y(I)
 PLOT 917        IF (Y(I) .GE. 99999.) YARRAY(I)=0.
 PLOT 918     40 CONTINUE
 PLOT 919        RETURN
 PLOT 920        END
END OF DATA
READY
```

Appendix A.7

COPY

```
DSNAME='IJL.COPY.PLI'
000010  /***** COPY - COPIES RECORDS FROM THE ACES DATA FILE          *****/
000020  /* T H I S   P R O G R A M   W I L L   C O P Y   R E C O R D S
000030     F R O M   T H E   A C E S   D A T A   F I L E   A N D   E I T H E R
000040     L I S T   T H E M   O N   T H E   L I N E   P R I N T E R   O R
000050     P U N C H   T H E M   O N T O   S O M E   O U T P U T   D E V I C E
000060  */
        COPY: PROC(PARM) OPTIONS(MAIN) REORDER;
000080   DCL SYSPRINT EXT FILE PRINT,
000090       SYSIN EXT FILE INPUT STREAM ENV(FB RECSIZE(80) BLKSIZE(800)),
000100       SYSUT1 EXT FILE RECORD KEYED DIRECT INPUT
000110              ENV(REGIONAL(1)),
000120       SYSUT2 EXT FILE STREAM OUTPUT ENV(FB RECSIZE(80) BLKSIZE(800)),
            (NO_DAYS,NO_RECORDS,I,CARD_LENGTH,J,J1,K,L,N,M) FIXED BIN(31),
            PARM CHAR(100) VARYING,
000140      1 INPUT,
000150        2 DATA(24) FLOAT DEC(6),
000160      (INPUT_CARD,FINAL_CARD) CHAR(1000) VARYING,
000170      C CHAR(20) VARYING, CARD_80 CHAR(80), LIST CHAR(8) VARYING,
000018      (DAY(8),RECORD(161)) FIXED BIN(31),
000190      IKEY FIXED DEC(5),
000200      (LENGTH,SUBSTR,INDEX) BUILTIN;
000210  /* O P E N I N G   O F   T H E   O U T P U T   F I L E
000220     ***** S Y S P R I N T *****   A N D   T H E   I N P U T
000230     F I L E   ***** S Y S I N *****  */
000240    OPEN FILE(SYSPRINT) LINESIZE(132); OPEN FILE(SYSIN);
000250  %PAGE;
000260  /*************************************************************************
000270   ************** I N F O R M A T I O N   S E C T I O N ******************
000280   *************************************************************************
000290   */
000300  /* T H E   F O L L O W I N G   T W O   P U T   S T A T E M E N T
000310     G I V E S   T H E   I N S T R U C T I O N S   O N   H O W
000320     T O   U S E   T H I S   P R O G R A M */
          IF INDEX(PARM,'NO')>0 THEN GO TO INPUT_SECTION;
000330    PUT FILE(SYSPRINT) EDIT
000340     ('THIS PROGRAM IS DESIGNED TO COPY ACES DATA FROM THE REGIONAL(1)',
000350      'DATASETS STORED ON DISK TO CARD IMAGES.  THE DATA IS PRESENTLY',
000360      'STORED ON A 2314 DISK WITH VOLUME NAME - UTACE1.',
000370      'TO USE THIS PROGRAM, IT IS NECESSARY TO INPUT CONTROL CARDS ON',
000380      'FILE - SYSIN.  THERE ARE THREE COMMANDS WHICH ARE ACCEPTED BY',
000390      'THIS PROGRAM.  THE COMMANDS ARE:',
000400      '   (1) - RECORDS=  FOLLOWED BY THE RECORD NUMBERS TO BE COPIED',
000410      '         SEPARATED BY COMMA''S WITH THE COMMAND STRING TERMI-',
000420      '         NATED BY A SEMICOLON.  IF ALL THE RECORDS ARE TO BE',
000430      '         COPIED, IT IS ONLY NECESSARY TO TYPE - RECORDS=ALL;',
000440      '   (2) - DAYS=  FOLLOWED BY THE DAYS TO BE COPIED SEPARATED BY',
000450      '         COMMA''S WITH THE COMMAND STRING TERMINATED BY A SEMI-',
000460      '         COLON.  IF ALL THE DAYS ARE TO BE COPIED, IT IS ONLY',
000470      '         NECESSARY TO TYPE - DAYS=ALL;')
000480      (PAGE,A,SKIP,A,SKIP,A,SKIP(2),A,SKIP,A,SKIP,A,SKIP(3),A,SKIP,A,
000490        SKIP,A,SKIP,A,SKIP(2),A,SKIP,A,SKIP,A,SKIP,A);
000500    PUT FILE(SYSPRINT) EDIT
```

```
000510      ('IN ADDITION TO THE TWO COMMANDS LISTED, BY TYPING - LISTROW; OR',
000520       'LISTCOL; - THE PROGRAM WILL LIST ON THE LINE PRINTER THE DATA',
000530       'BEING COPIED.  LISTROW PRODUCES THE LISTING IN ROW FORMAT AND',
000540       'LISTCOL PRODUCES THE LISTING IN COLUMN FORMAT.  LISTCOL IS',
000055       'LIMITED TO 10 RECORDS. LISTROW HAS A LIMIT OF 160 RECORDS.',
000560       'SPACES IN THE COMMAND STRING ARE IGNORED AND THE COMMANDS CAN',
000570       'BE ENTERED ON ANY NUMBER OF CARDS IN ANY ORDER.  INPUT CAN BE',
000580       'TERMINATED BY A BLANK CARD OR AN END-OF-FILE MARK.',
000059       'THE RECORD NUMBERS MUST BE BETWEEN 1 AND 160 AND THE DAY',
             'NUMBERS MUST BE BETWEEN 1 AND 7.A LIST OF RECORD NUMBERS AND',
   000       'THEIR MEANINGS CAN BE FOUND IN ACES DATA FILE LOGBOOK IN THE',
             'THE OFFICE OF - A.S. HOLMAN, BLDG 9102-1','')
000630      (SKIP(3),A,SKIP,A,SKIP,A,SKIP,A,SKIP,A,SKIP(2),A,SKIP,A,SKIP,A,
000640       SKIP(2),A,SKIP,A,SKIP,A,SKIP,A,SKIP,A);
000650  %PAGE;
        INPUT_SECTION:
000660  /*************************************************************************
000670   ********************** I N P U T   S E C T I O N **********************
000680   *************************************************************************
000690   */
000700  /* S E T   T H E   N U M B E R   O F   D A Y S   A N D   T H E
000710     N U M B E R   O F   R E C O R D S   T O   Z E R O */
000720    NO_DAYS,NO_RECORDS=0;
000730  /* S E T   T H E   E N D   O F   F I L E   C O N D I T I O N
000740     A N D   I N I T I A L   V A L U E   F O R   T H E
000750     I N P U T   C A R D */
000760    ON ENDFILE(SYSIN) GO TO ASHFILE001; INPUT_CARD='';
000770  /* T H E   F O L L O W I N G   1 0   C A R D S   P R O C E S S
000780     T H E   I N P U T   F I L E   S Y S I N */
000790    INPUT_LOOP: GET FILE(SYSIN) EDIT (CARD_80) (A(80));
000800    REVERT ENDFILE(SYSIN); ON ENDFILE(SYSIN) GO TO PROCESSING;
000810    INPUT_CARD=INPUT_CARD||CARD_80;
000820    IF CARD_80=(80) ' ' THEN GO TO PROCESSING; GO TO INPUT_LOOP;
000830    PROCESSING: REVERT ENDFILE(SYSIN); CLOSE FILE(SYSIN);
000840    CARD_LENGTH=LENGTH(INPUT_CARD); FINAL_CARD='';
000850    DO I=1 TO CARD_LENGTH;
000860       IF SUBSTR(INPUT_CARD,I,1)=' ' THEN J=1;
000870          ELSE FINAL_CARD=FINAL_CARD||SUBSTR(INPUT_CARD,I,1);
000880    END;
000890  /* T H E   F O L L O W I N G   1 9   C A R D S   C H E C K
000900     T H E   I N S T R U C T I O N   C A R D S   F O R   E R R O R S */
000910    I=INDEX(FINAL_CARD,'RECORDS=');
000920    IF I=0 THEN GO TO ASHFILE002;
000930    J=INDEX(FINAL_CARD,'DAYS=');
000940    IF J=0 THEN GO TO ASHFILE003;
000950    K=INDEX(FINAL_CARD,'LIST');
000960    IF K=0 THEN LIST='NOLIST';
000970       ELSE DO;
000980          K=INDEX(FINAL_CARD,'LISTROW');
000990          IF K>0 THEN LIST='LISTROW;'; ELSE LIST='LISTCOL;';
001000       END;
001010    L=INDEX(FINAL_CARD,LIST); CARD_LENGTH=LENGTH(FINAL_CARD);
001020    IF LENGTH(LIST)=8 THEN LIST=SUBSTR(LIST,1,7);
001030    IF L>0 THEN FINAL_CARD=SUBSTR(FINAL_CARD,1,L-1)||
001040       SUBSTR(FINAL_CARD,L+5,CARD_LENGTH-L-4);
001050       ELSE DO; IF K=0 THEN GO TO RETURN_1;
001060       FINAL_CARD=SUBSTR(FINAL_CARD,1,K-1)||
001070          SUBSTR(FINAL_CARD,K+4,CARD_LENGTH-K-3);
```

```
001080       GO TO ASHFILE004; END;
001090     RETURN_1: CARD_LENGTH=LENGTH(FINAL_CARD);
001100     DAY(*)=0; RECORD(*)=0; N,M=1;
001110   %PAGE;
001120   /*****************************************************************************
001130    ********** C A L C U L A T I O N   S E C T I O N ********************
001140    *****************************************************************************
001150    */
001160   /* T H E   F O L L O W I N G   1 4   C A R D S   D E T E R M I N E
001170      W H I C H   D A Y S   A R E   T O   B E   P R O C E S S E D */
001180     IF I<J THEN GO TO RECORDS;
001190     DAYS: IF M=0 THEN GO TO COPY; ON CONVERSION GO TO ASHFILE005;
001200     K=INDEX(FINAL_CARD,';'); L=INDEX(FINAL_CARD,',');
001210     IF K=6 THEN GO TO RECORDS1;
001220     IF (L=0)|(L>K) THEN DO; C=SUBSTR(FINAL_CARD,6,K-6); IF C='ALL'
001230     THEN DO; NO_DAYS=7; DO L=1 TO 7; DAY(L)=L; END; GO TO RECORDS1; END;
001240        NO_DAYS=NO_DAYS+1; DAY(NO_DAYS)=C; GO TO RECORDS1; END;
001250     C=SUBSTR(FINAL_CARD,6,L-6);
001260     FINAL_CARD=SUBSTR(FINAL_CARD,1,5)||
001270        SUBSTR(FINAL_CARD,L+1,CARD_LENGTH-L); CARD_LENGTH=CARD_LENGTH-L+5;
001280     NO_DAYS=NO_DAYS+1; DAY(NO_DAYS)=C;
001290     IF DAY(NO_DAYS)>7 THEN GO TO ASHFILE006;
001300     IF NO_DAYS>7 THEN GO TO ASHFILE007;
001310     GO TO DAYS;
001320   /* T H E   F O L L O W I N G   1 7   C A R D S   D E T E R M I N E
001330      W H I C H   R E C O R D S   A R E   T O   B E
001340      P R O C E S S E D */
001350     RECORDS1: K=INDEX(FINAL_CARD,';');
001360     FINAL_CARD=SUBSTR(FINAL_CARD,K+1,CARD_LENGTH-K);
001370     CARD_LENGTH=CARD_LENGTH-K; M=0;
001380     RECORDS: IF N=0 THEN GO TO COPY; ON CONVERSION GO TO ASHFILE008;
001390     K=INDEX(FINAL_CARD,';'); L=INDEX(FINAL_CARD,',');
001400     IF K=9 THEN GO TO DAYS1;
001410     IF (L=0)|(L>K) THEN DO; C=SUBSTR(FINAL_CARD,9,K-9); IF C='ALL'
             THEN DO; NO_RECORDS=160; DO L=1 TO 160; RECORD(L)=L; END;
             GO TO DAYS1;END; NO_RECORDS=NO_RECORDS+1; RECORD(NO_RECORDS)=C;
001440        GO TO DAYS1; END;
001450     C=SUBSTR(FINAL_CARD,9,L-9);
001460     FINAL_CARD=SUBSTR(FINAL_CARD,1,8)||
001470        SUBSTR(FINAL_CARD,L+1,CARD_LENGTH-L); CARD_LENGTH=CARD_LENGTH-L+8;
001480     NO_RECORDS=NO_RECORDS+1; RECORD(NO_RECORDS)=C;
000149     IF RECORD(NO_RECORDS)>160 THEN GO TO ASHFILE009;
000150     IF NO_RECORDS>160 THEN GO TO ASHFILE010;
001510     GO TO RECORDS;
001520     DAYS1: N=0; K=INDEX(FINAL_CARD,';');
001530     FINAL_CARD=SUBSTR(FINAL_CARD,K+1,CARD_LENGTH-K);
001540     CARD_LENGTH=LENGTH(FINAL_CARD); GO TO DAYS;
001550   /* T H E   F O L L O W I N G   5 0   C A R D S   O U T P U T
001560      T H E   R E C O R D S   T O   E I T H E R   T H E   L I N E
001570      P R I N T E R   O R   A   O U T P U T   F I L E   O R
001580      B O T H */
001590     COPY: IF NO_DAYS=0 THEN GO TO ASHFILE011;
001600     IF NO_RECORDS=0 THEN GO TO ASHFILE012;
001610     IF NO_RECORDS>10 THEN GO TO ASHFILE013;
001620   %PAGE;
```

```
001630  /*************************************************************************
001640   ***************** O U T P U T   S E C T I O N ***************************
001650   *************************************************************************
001660   */
001670    A: PUT FILE(SYSPRINT) EDIT
001680       ('THE FOLLOWING RECORDS WERE PROCESSED.') (PAGE,A);
001690    DO I=1 TO NO_DAYS;
001700       PUT FILE(SYSPRINT) EDIT ('RECORDS PROCESSED FOR DAY - ',DAY(I),'')
001710           (SKIP(2),A,F(2),SKIP(2),A);
001720       DO J=1 TO NO_RECORDS;
001730          IF J<NO_RECORDS THEN
001740          PUT FILE(SYSPRINT) EDIT (RECORD(J),',') (F(3),A(1));
001750          ELSE PUT FILE(SYSPRINT) EDIT (RECORD(J)) (F(3));
001760       END;
001770    END; OPEN FILE(SYSUT1); OPEN FILE(SYSUT2);
          IKEY=0; READ FILE(SYSUT1) INTO (INPUT) KEY(IKEY);
          PUT FILE(SYSUT2) EDIT (DATA(2),'/',DATA(3),'/',DATA(1),'')
                (F(2),A(1),F(2),A(1),F(2),SKIP,A);
001780    IF LIST='LISTCOL' THEN PUT FILE(SYSPRINT) EDIT
001790       ('*****  L I S T I N G   O F   R E C O R D S  *****','DAY',
001800        'HOUR',(RECORD(J) DO J=1 TO NO_RECORDS),'')
001810       (PAGE,X(41),A,SKIP(2),A,X(2),A,X(3),
001820        (NO_RECORDS) (X(5),F(3),X(5)),SKIP,A);
001830    DO I=1 TO NO_DAYS;
000184       K=(DAY(I)-1)*160;
001850       IF LIST='LISTROW' THEN PUT FILE(SYSPRINT) EDIT
001860          ('THE FOLLOWING OUTPUT IS FOR DAY - ',DAY(I))
001870          (PAGE,A,F(3));
001880       DO J=1 TO NO_RECORDS;
001890          IKEY=K+RECORD(J);
001900          READ FILE(SYSUT1) INTO (INPUT) KEY(IKEY);
001910          PUT FILE(SYSUT2) EDIT (DATA) (F(10,4));
001920          IF LIST='LISTROW' THEN PUT FILE(SYSPRINT) EDIT
001930             ('THE FOLLOWING DATA IS FOR RECORD - ',RECORD(J),DATA)
001940             (SKIP(3),A,F(3),SKIP(2),24 F(11,4));
001950       END;
001960       IF LIST='LISTCOL' THEN DO;
002020          DO J=1 TO 24;
001970          ON ENDPAGE(SYSPRINT) BEGIN; PUT FILE(SYSPRINT) EDIT
001980             ('*****  L I S T I N G   O F   R E C O R D S  *****','DAY',
                    'HOUR',(RECORD(J1) DO J1=1 TO NO_RECORDS),'')
002000             (PAGE,X(41),A,SKIP(2),A,X(2),A,X(3),
002010              (NO_RECORDS) (X(5),F(3),X(5)),SKIP,A); END;
                   PUT FILE(SYSPRINT) EDIT (DAY(I),J,'')
                          (SKIP,F(2),F(6),X(4),A);
002040             DO L=1 TO NO_RECORDS;
002050                IKEY=K+RECORD(L);
002060                READ FILE(SYSUT1) INTO (INPUT) KEY(IKEY);
002070                PUT FILE(SYSPRINT) EDIT (DATA(J)) (X(1),E(11,4,5));
002080       END; END; END;
002090    END;
002100    PUT FILE(SYSPRINT) EDIT ('COPY HAS BEEN COMPLETED.','')
002110      (PAGE,A,SKIP,A);
002120    CLOSE FILE(SYSPRINT); CLOSE FILE(SYSUT1); CLOSE FILE(SYSUT2);
```

```
002130    GO TO END_OF_COPY;
002140  %PAGE;
002150  /*************************************************************************
002160   ******************* E R R O R   S E C T I O N ***************************
002170   *************************************************************************
002180   */
002190  /* T H E   R E M A I N D E R   O F   T H E   P R O G R A M
002200     C O N T A I N S   T H E   E R R O R   M E S S A G E S   F O R
002210     T H E   P R O G R A M */
002220    ASHFILE001:
002230    PUT FILE(SYSPRINT) EDIT
002240      ('ASHFILE001 - THERE IS NO DATA ON INPUT FILE SYSIN.  RESUBMIT',
002250       '             JOB INCLUDING THE CONTROL CARDS.')
002260      (PAGE,A,SKIP,A); CLOSE FILE(SYSIN); CLOSE FILE(SYSPRINT);
002270    GO TO END_OF_COPY;
002280    ASHFILE002:
002290    PUT FILE(SYSPRINT) EDIT
002300      ('ASHFILE002 - THERE WAS NO RECORDS= CONTROL CARD ON INPUT FILE',
002310       '             SYSIN.  RESUBMIT JOB INCLUDING THE PROPER CONTROL',
002320       '             CARDS.') (PAGE,A,SKIP,A,SKIP,A);
002330    CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002340    ASHFILE003:
002350    PUT FILE(SYSPRINT) EDIT
002360      ('ASHFILE003 - THERE WAS NO DAYS= CONTROL CARD ON INPUT FILE',
002370       '             SYSIN.  RESUBMIT JOB INCLUDING THE PROPER CONTROL',
002380       '             CARDS.') (PAGE,A,SKIP,A,SKIP,A);
002390    CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002400    ASHFILE004:
002410    PUT FILE(SYSPRINT) EDIT
002420      ('ASHFILE004 - SEMICOLON ASSUMED AFTER LIST',
002430       '             PROGRAM IS CONTINUING.','')
002440      (SKIP(2),A,SKIP,A,SKIP,A);
002450    GO TO RETURN_1;
002460    ASHFILE005:
002470    PUT FILE(SYSPRINT) EDIT
002480      ('ASHFILE005 - STRING - ',C,' NOT RECOGNIZED AS A DAY',
002490       '             STRING HAS BEEN IGNORED.','')
002500      (SKIP(2),A,SKIP,A,SKIP,A); GO TO DAYS;
002510    ASHFILE006:
002520    PUT FILE(SYSPRINT) EDIT
002530      ('ASHFILE006 - DAY NUMBER -',DAY(NO_DAYS),' IS GREATER THAN 7.',
002540       '             CORRECT CONTROL CARDS AND RESUBMIT JOB.')
002550      (PAGE,A,F(10),A,SKIP,A); CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002560    ASHFILE007:
002570    PUT FILE(SYSPRINT) EDIT
002580      ('ASHFILE007 - NUMBER OF DAYS EXCEEDS 7. CORRECT CONTROL CARDS',
002590       '             AND RESUBMIT JOB.') (PAGE,A,SKIP,A);
002600    CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002610    ASHFILE008:
002620    PUT FILE(SYSPRINT) EDIT
002630      ('ASHFILE008 - STRING -',C,' NOT RECOGNIZED AS A RECORD NUMBER',
002640       '             STRING HAS BEEN IGNORED.','')
002650      (SKIP(2),A,A,A,SKIP,A,SKIP,A); GO TO RECORDS;
002660    ASHFILE009:
002670    PUT FILE(SYSPRINT) EDIT
002680      ('ASHFILE009 - RECORD NUMBER - ',RECORD(NO_RECORDS),'IS GREATER',
000269       'THAN 160.',
```

```
002700       '             CORRECT CONTROL CARDS AND RESUBMIT JOB.')
002710      (PAGE,A,F(10),A,A,SKIP,A); CLOSE FILE(SYSPRINT);
002720    GO TO END_OF_COPY;
002730    ASHFILE010:
002740    PUT FILE(SYSPRINT) EDIT
000275      ('ASHFILE010 - NUMBER OF RECORDS EXCEEDS 160.  CORRECT CONTROL',
002760       '             CARDS AND RESUBMIT JOB.') (PAGE,A,SKIP,A);
002770    CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002780    ASHFILE011:
002790    PUT FILE(SYSPRINT) EDIT
002800      ('ASHFILE011 - NO DAYS WERE FOUND TO PROCESS.  CORRECT CONTROL',
002810       '             CARDS AND RESUBMIT JOB.') (PAGE,A,SKIP,A);
002820    CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002830    ASHFILE012:
002840    PUT FILE(SYSPRINT) EDIT
002850      ('ASHFILE012 - NO RECORDS WERE FOUND TO PROCESS.  CORRECT CONTROL',
002860       '             CARDS AND RESUBMIT JOB.') (PAGE,A,SKIP,A);
002870    CLOSE FILE(SYSPRINT); GO TO END_OF_COPY;
002880    ASHFILE013:
002890    IF LIST='LISTROW' THEN GO TO A;
002900    IF LIST='NOLIST' THEN GO TO A;
002910    PUT FILE(SYSPRINT) EDIT
002920      ('ASHFILE013 - THE MAXIMUM NUMBER OF RECORDS FOR COLUMN TYPE',
002930       '             LISTING IS 11.  YOU HAVE ASKED FOR ',NO_RECORDS,
002940       ' RECORDS.',
002950       '             PROGRAM IS CONTINUING USINC ROW TYPE LISTING.','')
002960      (PAGE,A,SKIP,A,F(3),A,SKIP,A,SKIP,A); LIST='LISTROW';
002970      GO TO A;
002980  /*********************  E N D   O F   C O P Y  *********************/
002990  END_OF_COPY: END COPY;
END OF DATA
READY
```

Appendix A.8

ICE

```
DSNAME='IJL.ICE.FORT'
000010       REAL*4 ICE
             DIMENSION TEMP(365),HEAT(365),COOL(365),ICE(365),
000030      >          H(24),C(24),HW(24),OD(24),HR(24),SOL(24),Y(365)
000040       EQUIVALENCE (ICE(1),H(1)),(ICE(25),C(1)),(ICE(49),HW(1)),
000050      >            (ICE(73),OD(1)),(ICE(97),HR(1)),(ICE(121),SOL(1))
000060       REWIND 10
000070       DO 5 I=1,365
000080       HEAT(I)=0.
000090       COOL(I)=0.
000100     5 CONTINUE
000110       DO 20 I=1,365
000120       TAVG=0.
000130       READ(10) H,C,HW,OD,HR,SOL
000140       DO 10 K=1,24
000150       IF (H(K) .GT. 0.) H(K)=H(K)*1.2+.2*HW(K)
000160       HEAT(I)=H(K)+HEAT(I)
000170       IF (OD(K) .LT. 22.) C(K)=0.
000180       COOL(I)=C(K)+COOL(I)
000190       HEAT(I)=HW(K)+HEAT(I)
000200       TAVG=TAVG+OD(K)
000210    10 CONTINUE
000220       TEMP(I)=TAVG/24.
000230    20 CONTINUE
000240       DO 30 I=1,365
000250       ICE(I)=0.
000260       HEAT(I)=HEAT(I)/1000.
000270       COOL(I)=COOL(I)/1000.
000280    30 CONTINUE
000290       TBIN=32.
000300       TICE=0.
000310       DO 40 I=305,365
             TICE=TICE+(HEAT(I)*0.6296-COOL(I))*10.77-120.91
000330       IF (TICE .LT. 0.) TBIN=TBIN+(50.-TBIN)*TICE*9.7044E-8
000340       IF (TICE .LT. 0.) TICE=0.
000350       ICE(I)=TICE
000360    40 CONTINUE
000370       DO 50 I=1,304
000380       TICE=TICE+(HEAT(I)*0.6296-COOL(I))*10.77-181.82
000390       IF (TICE .LT. 0.) TBIN=TBIN+(50.-TBIN)*TICE*9.7044E-8
000400       IF (TICE .GT. 51000.) TICE=51000.
000410       IF (TICE .LT. 0.) TICE=0.
000420       ICE(I)=TICE
000430    50 CONTINUE
000440       DO 60 I=1,365
000450       ICE(I)=ICE(I)/1000.
000460       Y(I)=1.+FLOAT(I)*12./365.
000470    60 CONTINUE
              NDAYS=365
      C    IMARK IS THE FREQUENCY OF PLOTTING SYMBOLS
      C    NDAYS IS THE NUMBER OF DAYS IN A CALENDAR YEAR (JULIAN)
              IMARK=30
```

```
C     CHANGE SUBROUTINE FROM SEGMENT OF PROGRAMS
         CALL PLSUB1(HEAT,COOL,ICE,Y,TEMP,NDAYS,IMARK)
C         CALL PLSUB2(HEAT COOL,ICE,Y,TEMP,NDAYS,IMARK)
          STOP
          END
          SUBROUTINE PLSUB1(HEAT,COOL,ICE,Y,TEMP,NDAYS,IMARK)
          REAL*4 ICE
          DIMENSION HEAT(NDAYS),COOL(NDAYS),ICE(NDAYS),Y(NDAYS),
     >     TEMP(NDAYS),LEGND(200)
        DIMENSION WEEKH(53),WEEKC(53),WEEKT(53),TIME(53)
C   CHANGES TO PLSUB1 TO REDUCE SIZE OF PLOT
C                     TO SMOOTH OUT ERRATIC CURVES
          IM=1
          NOLD=1
          NWEEK=52
        NWP1=NWEEK+1
          ND=7
          NNEW=ND
C    ARRAYS OF AVERAGES TO BE PLOTTED
         WEEKH(1)=HEAT(1)
         WEEKC(1)=COOL(1)
          WEEKT(1)=TEMP(1)
         TIME(1)=Y(1)
        DO 35 IW=2,NWEEK
          SUMH=0.0
          SUMC=0.0
          SUMT=0.0
            IF(IW .GT. 2) NOLD=NNEW+1
            IF(IW .GT. 2) NNEW=NNEW+ND
C
              DO 25 ID=NOLD,NNEW
                SUMH=HEAT(ID)+SUMH
                SUMC=COOL(ID)+SUMC
                SUMT=TEMP(ID)+SUMT
 25           CONTINUE
C
            WEEKH(IW)=SUMH/ND
            WEEKC(IW)=SUMC/ND
            WEEKT(IW)=SUMT/ND
C
             TIME(IW)=Y(ID)
 40      FORMAT(3E12.4,F10.2)
 35       CONTINUE
         WEEKH(NWP1)=HEAT(365)
          WEEKC(NWP1)=COOL(365)
          WEEKT(NWP1)=TEMP(365)
          TIME(NWP1)=Y(365)
          WRITE(6,40) WEEKH,WEEKC,WEEKT,TIME
C    FACT IS MARKER SCALE
          FACT=2.0
000480       CALL CALCMP
000490       CALL BGNPL (1)
000500       CALL NEWPEN (1)
000510       CALL INCH30
000520       CALL PAGE (30.,37.)
000530       CALL COMPLX
000540       CALL BASALF ('L/CSTD')
000550       CALL MIXALF ('STANDARD')
```

```
000560       CALL HEIGHT (0.25)
             I=LINEST (LEGND,200,60)
000580       CALL LINES ('(O)UTSIDE (T)EMPERATURE (()(C)())$',LEGND,1)
             CALL LINES(
            <'(H)EATING AND (H)OT (W)ATER (L)OADS (()K(W)H())/(DAY)$',
            >            LEGND,2)
000610       CALL LINES ('(C)OOLING (L)OADS (()K(W)H())/(DAY)$',LEGND ,3)
000620       CALL LINES ('(I)CE (I)NVENTORY (()(M)T())$',LEGND,4)
000630       XL=XLEGND(LEGND,4)+0.4
000640       YL=YLEGND(LEGND,4)+0.40
000650       CALL RESET ('HEIGHT')
000660       CALL YINTAX
000670       CALL YAXANG (0.)
000680       CALL YTICKS (1)
000690       CALL XTICKS (1)
000700       CALL HEIGHT (0.30)
      C     SCALE DOWN THE PLOT BY 70%
             CALL BLOWUP(0.30)
000710       CALL TITLE ('(F)IRST (Y)EAR (P)LOT FOR (ACES)$',-100,
000720      >            'T(IME) (()M(ONTH)())$',100,
000730      >            'T(EMPERATURE) (()D(EG) C())$',100,30.,24.)
000740       CALL RESET ('HEIGHT')
000750       CALL HEIGHT (0.20)
000760       CALL BLNK1 (0.,XL,24.-YL,24.,2)
000770       CALL BASALF ('STANDARD')
000780       CALL MIXALF ('L/CSTD')
000790       CALL XMONGR (1,13,-20.,10.,50.)
000800       CALL BASALF('L/CSTD')
000810       CALL MIXALF('STANDARD')
      C    SMOOTH THE ERRATIC DATA CURVES FOR HEAT COOL TEMP
            CALL SCLPIC(FACT)
            CALL CURVE (TIME,WEEKT,NWP1,IM)
000830      XPOS=XPOSN (1.,-20.)
000840      YPOS=YPOSN (1.,-20.)
000850      CALL YGRAXS (0.,30.,210.,24.,
000860     >             '(B)UILDING (L)OADS (()K(W)H())/(DAY)$',-100,
000870     >             XPOS,YPOS)
000880      CALL NEWPEN (2)
              CALL SCLPIC(FACT)
   00         CALL CURVE (TIME,WEEKH,NWP1,IM)
000900       CALL NEWPEN (3)
             CALL SCLPIC(FACT)
   00        CALL CURVE (TIME,WEEKC,NWP1,IM)
000920       XPOS=XPOSN (13.,0.)
000930       YPOS=YPOSN (13.,0.)
000940       CALL NEWPEN (1)
000950       CALL YGRAXS (0.,10.,70.,24.,
000960      >             '(I)CE (I)NVENTORY (()(M)T())$',-100,
000970      >             XPOS,YPOS)
              CALL SCLPIC(FACT)
   000        CALL CURVE (Y,ICE,NDAYS,IMARK)
000990       CALL NEWPEN (1)
001000       CALL RLVEC (1.,10.,13.,10.,0)
001010       CALL RLVEC (1.,20.,13.,20.,0)
001020       CALL RLVEC (1.,30.,13.,30.,0)
001030       CALL RLVEC (1.,40.,13.,40.,0)
001040       CALL RLVEC (1.,50.,13.,50.,0)
001050       CALL RLVEC (1.,60.,13.,60.,0)
```

```
001060       CALL RLVEC (2.,0.,2.,70.,0)
001070       CALL RLVEC (3.,0.,3.,70.,0)
001080       CALL RLVEC (4.,0.,4.,70.,0)
001090       CALL RLVEC (5.,0.,5.,70.,0)
001100       CALL RLVEC (6.,0.,6.,70.,0)
001110       CALL RLVEC (7.,0.,7.,70.,0)
001120       CALL RLVEC (8.,0.,8.,70.,0)
001130       CALL RLVEC (9.,0.,9.,70.,0)
001140       CALL RLVEC (10.,0.,10.,70.,0)
001150       CALL RLVEC (11.,0.,11.,70.,0)
001160       CALL RLVEC (12.,0.,12.,70.,0)
001170       CALL RESET ('HEIGHT')
001180       CALL HEIGHT(0.25)
001190       CALL RESET ('BLNKS')
001200       YL=YL-0.2
001210       CALL LEGEND (LEGND,4,0.2,24.-YL)
001220       CALL FRAME
001230       CALL ENDPL (1)
001240       CALL DONEPL
000012       RETURN
001260       END
               SUBROUTINE PLSUB2(HEAT,COOL,ICE,Y,TEMP,NDAYS,IMARK)
               REAL*4 ICE
               DIMENSION HEAT(NDAYS),COOL(NDAYS),ICE(NDAYS),Y(NDAYS),
            >   TEMP(NDAYS),LEGND(200)
000480       CALL CALCMP
000490       CALL BGNPL (1)
000500       CALL NEWPEN (1)
000510       CALL INCH30
000520       CALL PAGE (30.,37.)
000530       CALL COMPLX
000540       CALL BASALF ('L/CSTD')
000550       CALL MIXALF ('STANDARD')
000560       CALL HEIGHT (0.25)
             I=LINEST (LEGND,200,60)
000580       CALL LINES ('(O)UTSIDE (T)EMPERATURE (()(C)())$',LEGND,1)
             CALL LINES(
            <'(H)EATING AND (H)OT (W)ATER (L)OADS (()K(W)H())/(DAY)$',
            >            LEGND,2)
000610       CALL LINES ('(C)OOLING (L)OADS (()K(W)H())/(DAY)$',LEGND ,3)
000620       CALL LINES ('(I)CE (I)NVENTORY (()(M)T())$',LEGND,4)
000630       XL=XLEGND(LEGND,4)+0.4
000640       YL=YLEGND(LEGND,4)+0.40
000650       CALL RESET ('HEIGHT')
000660       CALL YINTAX
000670       CALL YAXANG (0.)
000680       CALL YTICKS (1)
000690       CALL XTICKS (1)
000700       CALL HEIGHT (0.30)
000710       CALL TITLE ('(F)IRST (Y)EAR (P)LOT FOR (ACES)$',-100,
000720      >            'T(IME) (()M(ONTH)())$',100,
000730      >            'T(EMPERATURE) (()D(EG) C())$',100,30.,24.)
000740       CALL RESET ('HEIGHT')
000750       CALL HEIGHT (0.20)
000760       CALL BLNK1 (0.,XL,24.-YL,24.,2)
000770       CALL BASALF ('STANDARD')
000780       CALL MIXALF ('L/CSTD')
000790       CALL XMONGR (1,13,-20.,10.,50.)
```

```
000800        CALL BASALF('L/CSTD')
000810        CALL MIXALF('STANDARD')
   000        CALL CURVE (Y,TEMP,NDAYS,IMARK)
000830        XPOS=XPOSN (1.,-20.)
000840        YPOS=YPOSN (1.,-20.)
000850        CALL YGRAXS (0.,30.,210.,24.,
000860       >             '(B)UILDING (L)OADS (()K(W)H())/(DAY)$',-100,
000870       >             XPOS,YPOS)
000880        CALL NEWPEN (2)
   000        CALL CURVE (Y,HEAT,NDAYS,IMARK)
000900        CALL NEWPEN (3)
   000        CALL CURVE (Y,COOL,NDAYS,IMARK)
000920        XPOS=XPOSN (13.,0.)
000930        YPOS=YPOSN (13.,0.)
000940        CALL NEWPEN (1)
000950        CALL YGRAXS (0.,10.,70.,24.,
000960       >             '(I)CE (I)NVENTORY (()(M)T())$',-100,
000970       >             XPOS,YPOS)
   000        CALL CURVE (Y,ICE,NDAYS,IMARK)
000990        CALL NEWPEN (1)
001000        CALL RLVEC (1.,10.,13.,10.,0)
001010        CALL RLVEC (1.,20.,13.,20.,0)
001020        CALL RLVEC (1.,30.,13.,30.,0)
001030        CALL RLVEC (1.,40.,13.,40.,0)
001040        CALL RLVEC (1.,50.,13.,50.,0)
001050        CALL RLVEC (1.,60.,13.,60.,0)
001060        CALL RLVEC (2.,0.,2.,70.,0)
001070        CALL RLVEC (3.,0.,3.,70.,0)
001080        CALL RLVEC (4.,0.,4.,70.,0)
001090        CALL RLVEC (5.,0.,5.,70.,0)
001100        CALL RLVEC (6.,0.,6.,70.,0)
001110        CALL RLVEC (7.,0.,7.,70.,0)
001120        CALL RLVEC (8.,0.,8.,70.,0)
001130        CALL RLVEC (9.,0.,9.,70.,0)
001140        CALL RLVEC (10.,0.,10.,70.,0)
001150        CALL RLVEC (11.,0.,11.,70.,0)
001160        CALL RLVEC (12.,0.,12.,70.,0)
001170        CALL RESET ('HEIGHT')
001180        CALL HEIGHT(0.25)
001190        CALL RESET ('BLNKS')
001200        YL=YL-0.2
001210        CALL LEGEND (LEGND,4,0.2,24.-YL)
001220        CALL FRAME
001230        CALL ENDPL (1)
001240        CALL DONEPL
000012        RETURN
001260        END
END OF DATA
READY
```

Appendix A.9

UTTVA

```
DSNAME='IJL.UTTVA.PLI'
000010  UTTVA: PROC OPTIONS(MAIN);
000020    DCL SYSPRINT EXT FILE PRINT,
000030        ACEDATA EXT FILE RECORD DIRECT INPUT KEYED ENV(REGIONAL(1)),
000040       UTTAPE EXT FILE,
000050       TVATAPE EXT FILE,
000060       IKEY FIXED DEC(5),
000070       (I,J) FIXED BINARY(15),
000080       1 INPUT,
000090         2 DATA(24) FLOAT DEC(6);
000100      IKEY=0; READ FILE(ACEDATA) INTO (INPUT) KEY (IKEY);
000110      PUT FILE(UTTAPE) EDIT (DATA) (24 F(20,8));
000120      PUT FILE(TVATAPE) EDIT (DATA) (24 F(20,8));
000130      DO I=1 TO 160;
000140         DO J=1 TO 7;
000150           IKEY=(J-1)*160+I; READ FILE(ACEDATA) INTO (INPUT) KEY (IKEY);
000160              PUT FILE(UTTAPE) EDIT (DATA) (24 F(20,8));
000170            PUT FILE(TVATAPE) EDIT (DATA) (24 F(20,8));
000180         END;
000190      END;
000200      PUT FILE(SYSPRINT) EDIT ('COPY HAS BEEN COMPLETED TO UT AND ',
000210                'TVA TAPES.','') (PAGE,A,A,SKIP,A);
000220  END UTTVA;
END OF DATA
READY
```

Appendix B

JCL LISTINGS

```
MEMBER NAME  COPY
//IJLACOPY JOB (15936),'9102-1 BLEDSOE',MSGCLASS=T
/*ROUTE PRINT REMOTE4
//*CLASS CPU91=7S,IO=2,REGION=80K
//COPYACES EXEC PGM=ACESCOPY,PARM='ISA(8K)/NO',REGION=80K
//STEPLIB  DD DSN=PLI.LINKLIB,DISP=SHR
//         DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=T.IJL05936.D78324,DISP=SHR
//SYSUT2   DD DUMMY
//SYSIN    DD *
 RECORDS=114,31,32,34,36,37,38,39,40; DAYS=ALL; LISTCOL;
/*
//

MEMBER NAME  FILESPDA
//IJLSPDA1 JOB (15936),'9102-1 J BLEDSOE',MSGCLASS=T
/*ROUTE PRINT REMOTE4
//*CLASS CPU91=10S,IO=2.0
//SPDASCRT EXEC SPDASCR
//USERSCR.SYSIN DD *
          T.IJL05936.D78324
/*
//ACESFILE EXEC PGM=ACESFILE,PARM='ISA(50K)/D78324,NOSORT',
//         REGION=270K
//STEPLIB  DD DSN=PLI.LINKLIB,DISP=SHR
//         DD DSN=IJL.PRIVATE.LIBRARY,
//            DISP=(OLD,KEEP)
//         DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTLIB  DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//PRINT    DD SYSOUT=A
//PRINTER  DD SYSOUT=A
//ACESOUT  DD DSN=T.IJL05936.D78324,UNIT=SPDA,DISP=(NEW,CATLG),
//            DCB=(RECFM=F,BLKSIZE=96),SPACE=(TRK,(20,20),RLSE)
//SORTOUT  DD UNIT=SYSDA,SPACE=(CYL,(4,1)),
//            DCB=(RECFM=FB,LRECL=960,BLKSIZE=7680)
//SORTIN   DD UNIT=SYSDA,SPACE=(CYL,(4,1)),
//            DCB=(RECFM=FB,LRECL=960,BLKSIZE=7680)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4,1))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(4,1))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(4,1))
//ACESIN   DD DSN=T.IJL05936.C78323,DISP=SHR
//
```

```
MEMBER NAME  PLOTONLY
//IJLPLOT2 JOB (15936),'9102-1  BLEDSOE',MSGCLASS=T
//*PLOT  TYPE=CAL925,PAPE=600,COPI=1,NUMB=9,INK=(BLACK/L,RED/L,BLUE/L)
/*ROUTE PRINT REMOTE4
//*CLASS CPU91=60S,IO=10,REGION=270,SPECIAL=TAPE
//PASSPLOT EXEC PGM=SUMTEST,PARM='ISA(4K,30K,4)/BOTH,ZERO',
//         REGION=270K
//STEPLIB  DD DSN=PLI.LINKLIB,DISP=SHR
//         DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSPRT1  DD SYSOUT=A
//SYSPRT2  DD SYSOUT=A
//SYSPRT3  DD SYSOUT=A
//TOTAL    DD DSN=T.IJL05936.DATATOTL,DISP=SHR
//*           VOL=SER=UTACE1,UNIT=2314
//ACEDATA  DD DSN=T.IJL05936.D80014,DISP=SHR
//*           VOL=SER=UTACE2,UNIT=2314
//PLOTS    DD UNIT=SYSDA,DSN=&&PLOTDATA,SPACE=(TRK,(50,50)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(NEW,PASS)
//PLOTACES EXEC PGM=ACESPLOT,COND=(7,LT,PASSPLOT),REGION=270K
//STEPLIB  DD DSN=IJL.PRIVATE.LIBRARY,
//            DISP=SHR
//FT06F001 DD SYSOUT=A
//FT10F001 DD DSN=&&PLOTDATA,DISP=(OLD,DELETE)
//FT54F001 DD DDNAME=PLOTTAPE
//PLOTTAPE DD UNIT=TAPE9,VOL=SER=10,LABEL=(,NL),DISP=OLD,
//            DCB=(RECFM=VS,LRECL=364,BLKSIZE=368)
//FT05F001 DD DUMMY
//PASSSOLR EXEC PGM=ACESCOPY,PARM='ISA(8K)/NO',REGION=80K
//STEPLIB  DD DSN=PLI.LINKLIB,DISP=SHR
//         DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//SYSPRINT DD DUMMY
//SYSUT1   DD DSN=ACES.DATA.D80014,DISP=(OLD,KEEP),
//            UNIT=2314,VOL=SER=UTACE2
//SYSUT2   DD UNIT=SYSDA,SPACE=(CYL,(4,1)),DSN=&&SOLRDATA,
//            DISP=(NEW,PASS)
//SYSIN    DD *
  RECORDS=114,31,32,34,36,37,38,39,40; DAYS=ALL; LISTCOL;
/*
//SOLARSUM EXEC PGM=ACESSOLR,REGION=80K
//STEPLIB  DD DSN=HOLMAN.PRIVATE.LIBRARY,UNIT=2314,VOL=SER=UTACE1,
//            DISP=(OLD,KEEP)
//FT06F001 DD SYSOUT=A
//FT01F001 DD DSN=&&SOLRDATA,DISP=(OLD,DELETE)
//
```

```
MEMBER NAME  PLOTYEAR
//IJLPLYR JOB (15936),'Y12 9102-1 J BLEDSOE',MSGCLASS=T
/*ROUTE  PRINT REMOTE4
//*PLOT  TYPE=CAL925,COPI=1,PAPE=500,NUMB=1,INK=(BLACK/L,RED/L,BLUE/L)
//*CLASS CPU91=10S,IO=1,REGION=270,SPECIAL=TAPE
//PLOT     EXEC PGM=PLOTYEAR,REGION=270K
//STEPLIB  DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//FT06F001 DD SYSOUT=A
//FT10F001 DD DSN=IJL.ACESDEMO.YEARDATA,DISP=SHR
//FT54F001 DD DDNAME=PLOTTAPE
//PLOTTAPE DD UNIT=TAPE9,VOL=SER=10,LABEL=(,NL),DISP=OLD,
//            DCB=(RECFM=VS,LRECL=364,BLKSIZE=368)
//FT05F001 DD DUMMY
//
```

```
MEMBER NAME  FILEDISK
//IJLFILE1 JOB (15936),'Y12 9102-1 J BLEDSOE',MSGCLASS=T
//*NOTES  TAG TAPES TO J BLEDSOE 9102-1
//*PLOT  TYPE=CAL925,PAPE=600,COPI=1,NUMB=9,INK=(BLACK/L,RED/L,BLUE/L)
//*CLASS CPU91=2M,IO=12,REGION=270,SPECIAL=TAPE
//SPDASCRT EXEC SPDASCR
//USERSCR.SYSIN DD *
          T.IJL05936.D80098
/*
//ACESFILE EXEC PGM=ACESFILE,PARM='ISA(16K)/D80098,NOSORT',
//         REGION=270K
//STEPLIB  DD DSN=PLI.LINKLIB,DISP=SHR
//         DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//         DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTLIB  DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=A
//PRINT    DD SYSOUT=A
//PRINTER  DD SYSOUT=A
//ACESOUT  DD DSN=T.IJL05936.D80098,UNIT=SPDA,DISP=(NEW,CATLG),
//            DCB=(RECFM=F,BLKSIZE=96),SPACE=(TRK,(20,20),RLSE)
//SORTOUT  DD UNIT=SYSDA,SPACE=(CYL,(4,1)),
//            DCB=(RECFM=FB,LRECL=960,BLKSIZE=6720)
//SORTIN   DD UNIT=SYSDA,SPACE=(CYL,(4,1)),
//            DCB=(RECFM=FB,LRECL=960,BLKSIZE=6720)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4,1))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(4,1))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(4,1))
//ACESIN   DD DSN=T.IJL05936.C80084,DISP=SHR
//         DD DSN=T.IJL05936.C80098,DISP=SHR
//DD1      DD UNIT=2314,VOL=SER=UTACE2,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
 SCRATCH DSNAME=ACES.DATA.D80098,VOL=2314=UTACE2
/*
//COPYACES EXEC PGM=IEBGENER,REGION=64K
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=T.IJL05936.D80098,DISP=SHR
//SYSUT2   DD DSN=ACES.DATA.D80098,UNIT=2314,VOL=SER=UTACE2,
//            SPACE=(TRK,(10,10),RLSE),DISP=(NEW,KEEP),
//            DCB=(RECFM=F,LRECL=96,BLKSIZE=96)
//SYSIN    DD DUMMY
//LISTCAT  EXEC PGM=IEHLIST,REGION=64K
//SYSPRINT DD SYSOUT=A
//DD2      DD UNIT=2314,VOL=SER=UTACE2,DISP=OLD
//SYSIN    DD *
 LISTVTOC FORMAT,VOL=2314=UTACE2
/*
//PASSPLOT EXEC PGM=SUMTEST,PARM='ISA(4K,30K,4)/BOTH',
//         REGION=270K
//STEPLIB  DD DSN=PLI.LINKLIB,DISP=SHR
//         DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSPRT1  DD SYSOUT=A
//SYSPRT2  DD SYSOUT=A
//SYSPRT3  DD SYSOUT=A
//TOTAL    DD DSN=ACES.DATA.TOTALS,DISP=(OLD,KEEP),
//            VOL=SER=UTACE1,UNIT=2314
//ACEDATA  DD DSN=ACES.DATA.D80098,DISP=(OLD,KEEP),
//            VOL=SER=UTACE2,UNIT=2314
```

```
MEMBER NAME  FILEDISK
//PLOTS     DD UNIT=SYSDA,DSN=&&PLOTDATA,SPACE=(TRK,(50,50)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=(NEW,PASS)
//PLOTACES  EXEC PGM=ACESPLOT,COND=((7,LT,ACESFILE),(7,LT,PASSPLOT)),
//          REGION=270K
//STEPLIB   DD DSN=IJL.PRIVATE.LIBRARY,
//             DISP=SHR
//FT06F001  DD SYSOUT=A
//FT10F001  DD DSN=&&PLOTDATA,DISP=(OLD,DELETE)
//FT54F001  DD DDNAME=PLOTTAPE
//PLOTTAPE  DD UNIT=TAPE9,VOL=SER=10,LABEL=(,NL),DISP=OLD,
//             DCB=(RECFM=VS,LRECL=364,BLKSIZE=368)
//FT05F001  DD DUMMY
//PASSSOLR  EXEC PGM=ACESCOPY,PARM='ISA(8K)/NO',REGION=270K
//STEPLIB   DD DSN=PLI.LINKLIB,DISP=SHR
//          DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//SYSPRINT  DD DUMMY
//SYSUT1    DD DSN=ACES.DATA.D80098,DISP=(OLD,KEEP),
//             UNIT=2314,VOL=SER=UTACE2
//SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(4,1)),DSN=&&SOLRDATA,
//             DISP=(NEW,PASS)
//SYSIN     DD *
  RECORDS=114,31,32,34,36,37,38,39,40; DAYS=ALL; LISTCOL;
/*
//SOLARSUM  EXEC PGM=ACESSOLR,REGION=270K
//STEPLIB   DD DSN=HOLMAN.PRIVATE.LIBRARY,UNIT=2314,VOL=SER=UTACE1,
//             DISP=(OLD,KEEP)
//FT06F001  DD SYSOUT=A
//FT01F001  DD DSN=&&SOLRDATA,DISP=(OLD,DELETE)
//TAPECOPY  EXEC PGM=UTTVA,PARM='ISA(20K)',REGION=64K
//STEPLIB   DD DSN=PLI.LINKLIB,DISP=SHR
//          DD DSN=IJL.PRIVATE.LIBRARY,DISP=SHR
//SYSPRINT  DD SYSOUT=A
//ACEDATA   DD DSN=ACES.DATA.D80098,DISP=(OLD,KEEP),
//             VOL=SER=UTACE2,UNIT=2314
//TVATAPE   DD UNIT=TAPE9,VOL=SER=TAG160,LABEL=(,SL),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//             DSN=ORNL.TVATAPE.DATA,DISP=(NEW,PASS)
//UTTAPE    DD UNIT=TAPE9,VOL=SER=TAG122,LABEL=(,SL),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//             DSN=ORNL.UTTAPE.DATA,DISP=(NEW,PASS)
//
```

Appendix C

TSO COMMAND PROCEDURES

```
MEMBER NAME  ACESFILE
PROC 0 EXP M(91) UT(<) TVA(<) DSNAME(<) NODSN(1) DSN3(<) DSN2(<) DSN1(<)
IF (&DSNAME. EQ <) GO TO LABEL 100
GO TO LABEL 200
LABEL 100
TELL 'THE FOLLOWING IS A SHORT HELP FILE ON THE USE OF THIS CLIST.'
TELL ' '
TELL 'THIS CLIST IS INVOKED BY ENTERING -'
TELL ' '
TELL 'EX CMDPROC(ACESFILE) '' OPTIONS '' '
TELL ' '
TELL 'WHERE OPTIONS ARE -'
TELL '         DSNAME() - NAME OF NEW DATASET TO BE CREATED ON BOTH DISK'
TELL '                    UTACE2 AND ON SPDA.'
TELL '         NODSN()  - NO OF OLD DATASETS TO BE CONCATANATED TO PRO-'
TELL '                    DUCE THE NEW DATASET. DEFAULT IS 1.'
TELL '         DSN1()   - LAST INDEX STRUCTURE OF FIRST DATASET TO BE'
TELL '                    CONCATANATED. CLIST ASSUMES HIGH LEVEL INDEX'
TELL '                    FOR DSN1,DSN2, AND DSN3 OF - T.IJL05936.'
TELL '         DSN2()   - LAST INDEX STRUCTURE OF SECOND DATASET TO BE'
TELL '                    CONCATANATED.'
TELL '         DSN3()   - LAST INDEX STRUCTURE OF THIRD DATASET TO BE'
TELL '                    CONCATANATED.'
TELL '         EXP      - ENTER EXP IF THE DATA INVOLVED WAS PART OF A '
TELL '                    SPECIAL EXPERIMENT.'
TELL '         UT()     - THE NUMBER OF THE UT TAG TAPE '
TELL '         TVA()    - THE NUMBER OF THE TVA TAG TAPE  '
TELL '         M()      - CPU MODEL NUMBER FOR JOB DEFAULT IS 91'
TELL ' '
TELL 'REQUIRED - DSNAME(DSN0) DSN1(DSN1) UT(TAG???) TVA(TAG???)'
TELL 'DEFAULTS - NODSN(1)'
TELL ' '
TELL 'NOTES - NODSN(NO) MUST BE EQUAL TO THE NUMBER OF DATASETS THAT'
TELL '        ARE TO BE CONCATANATED.  THE MAXIMUM NUMBER OF CONCAT-'
TELL '        ANATED DATASETS IS 3.'
TELL ' '
TELL 'REEXECUE CLIST USING THESE INSTRUCTIONS'
GO TO LABEL 9999
LABEL 200
IF (&TVA. NE <) GO TO LABEL 220
TELL 'YOU MUST INCLUDE THE NUMBER OF THE TVA TAPE.'
GO TO LABEL 9999
LABEL 220
IF (&UT. NE <) GO TO LABEL 240
TELL 'YOU MUST INCLUDE THE NUMBER OF THE UT TAPE.'
GO TO LABEL 9999
LABEL 240
TELL 'DATASET - T.IJL05936.&DSNAME. IS BEING SUBMITTED TO SPDA.'
TELL 'DATASET - ACES.DATA.&DSNAME IS BEING SUBMITTED TO UTACE2.'
IF (&NODSN. EQ 1) GO TO LABEL 300
IF (&NODSN. EQ 2) GO TO LABEL 400
IF (&NODSN. EQ 3) GO TO LABEL 500
TELL 'NO OF DATASETS IS LIMITED TO 3.'
GO TO LABEL 9999
LABEL 300
IF (&DSN1. NE <) GO TO LABEL 350
TELL 'YOU MUST INCLUDE THE DSN FOR DSN1.'
GO TO LABEL 9999
LABEL 350
```

```
MEMBER NAME  ACESFILE
QEDIT JOB(FILEDISK) CNTL
C 90 /DDDDDD/&DSNAME.,&EXP./
C 70 910 /DDDDDD/&DSNAME./ALL
C 270 280 /DDDNAME1/&DSN1./ALL
C 940 /TAPTVA/&TVA./
C 970 /TAPEUT/&UT./
END S
SUB JOB(FILEDISK) MODEL(&M.) NOSEQ
QEDIT JOB(FILEDISK) CNTL
C 90 /&DSNAME.,&EXP./DDDDDD/
C 70 910 /&DSNAME./DDDDDD/ALL
C 270 280 /&DSN1./DDDNAME1/ALL
C 940 /&TVA./TAPTVA/
C 970 /&UT./TAPEUT/
END S
GO TO LABEL 600
LABEL 400
IF (&DSN1. NE <) GO TO LABEL 410
TELL 'YOU MUST INCLUDE THE DSN FOR DSN1.'
GO TO LABEL 9999
LABEL 410
IF (&DSN2. NE <) GO TO LABEL 450
TELL 'YOU MUST INCLUDE THE DSN FOR DSN2.'
GO TO LABEL 9999
LABEL 450
QEDIT JOB(FILEDISK) CNTL
C 90 /DDDDDD/&DSNAME.,&EXP./
C 70 910 /DDDDDD/&DSNAME./ALL
C 270 280 /DDDNAME1/&DSN1./ALL
280 //          DD DSN=T.IJL05936.&DSN2.,DISP=SHR
C 940 /TAPTVA/&TVA./
C 970 /TAPEUT/&UT./
END S
SUB JOB(FILEDISK) MODEL(&M.) NOSEQ
QEDIT JOB(FILEDISK) CNTL
C 90 /&DSNAME.,&EXP./DDDDDD/
C 70 910 /&DSNAME./DDDDDD/ALL
C 270 280 /&DSN1./DDDNAME1/ALL
D 280
C 940 /&TVA./TAPTVA/
C 970 /&UT./TAPEUT/
END S
GO TO LABEL 600
LABEL 500
IF (&DSN1. NE <) GO TO LABEL 510
TELL 'YOU MUST INCLUDE THE DSN FOR DSN1.'
GO TO LABEL 9999
LABEL 510
IF (&DSN2. NE <) GO TO LABEL 520
TELL 'YOU MUST INCLUDE THE DSN FOR DSN2.'
GO TO LABEL 9999
LABEL 520
IF (&DSN3. NE <) GO TO LABEL 550
TELL 'YOU MUST INCLUDE THE DSN FOR DSN3.'
GO TO LABEL 9999
LABEL 550
QEDIT JOB(FILEDISK) CNTL
C 90 /DDDDDD/&DSNAME.,&EXP./
```

```
MEMBER NAME  ACESFILE
C 70 910 /DDDDDD/&DSNAME./ALL
C 270 280 /DDDNAME1/&DSN1./ALL
280 //          DD DSN=T.IJL05936.&DSN2.,DISP=SHR
272 //          DD DSN=T.IJL05936.&DSN3.,DISP=SHR
C 930 /TAPTVA/&TVA./
C 970 /TAPEUT/&UT./
END S
SUB JOB(FILEDISK) MODEL(&M.) NOSEQ
QED JOB(FILEDISK) CNTL
C 90 /&DSNAME.,&EXP./DDDDDD/
C 70 910 /&DSNAME./DDDDDD/ALL
C 270 280 /&DSN1./DDDNAME1/ALL
D 280 272
C 930 /&TVA./TAPTVA/
C 970 /&UT./TAPEUT/
END S
LABEL 600
COMP JOB.CNTL
SH DA(JOB.CNTL) REL
LABEL 9999
TELL ' '
TELL 'THE PROGRAM IS TERMINATING'
ENT
```

```
MEMBER NAME  YEARFILE
 PROC 1 DSNAME DATASET(VERYNEW)
 STEPLIB 'PLI.LINKLIB'
 $FREE FI(SYSPRINT SYSIN SYSUT1 SYSUT2)
 FREE ATTRLIST(LIST1)
 ATTRIB LIST1 RECFM(V B A) LRECL(137) BLKSIZE(1100)
 ALLOC FI(SYSPRINT) NEW BLOCK(1100) USING(LIST1) SP(20 20)
 ALLOC FI(SYSIN) DA(YEARLY.FILE) SHR
 ALLOC FI(SYSUT1) DA('T.IJL05936.&DSNAME.') SHR
 ALLOC FI(SYSUT2) DA(ACESDATA.&DATASET.) MOD
 CALL 'IJL.PRIVATE.LIBRARY(ACESCOPY)' 'ISA(8K)/NO'
 FREE FI(SYSPRINT SYSIN SYSUT1 SYSUT2)
 STEPLIB
 END
```

```
MEMBER NAME  PROCYEAR
PROC 0
$FREE FI(FT05F001 FT06F001 FT10F001 FT11F001 FT12F001 FT10F002)
ALLOC FI(FT05F001) DA(*)
ALLOC FI(FT06F001) DA(*)
ALLOC FI(FT11F001) DA(ACESDEMO.YEARDATA) SHR
ALLOC FI(FT10F001) DA(ACESDATA.VERYNEW) SHR
ALLOC FI(FT10F002) DA(ACESDATA.VERYOLD) SHR
CONCAT (FT10F001 FT10F002)
FREE ATTRLIST(LIST1)
ATTRIB LIST1 RECFM(V B S) LRECL(484) BLKSIZE(500)
ALLOC FI(FT12F001) NEW BLOCK(500) USING(LIST1) SP(100 100)
CALL 'IJL.PRIVATE.LIBRARY(TAPEFIX)'
FREE FI(FT10F001 FT10F002 FT11F001 FT12F001 FT05F001 FT06F001)
FREE ATTRLIST(LIST1)
END

MEMBER NAME  MONTHLY
 PROC 0
 $FREE FI(FT10F001 FT06F001 FT05F001)
 ALLOC FI(FT06F001) DA(*)
 ALLOC FI(FT05F001) DA(*)
 ALLOC FI(FT10F001) DA(ACESDEMO.YEARDATA) SHR
 CALL 'IJL.PRIVATE.LIBRARY(MONTHLY)'
 FREE FI(FT10F001 FT06F001 FT05F001)
 END
```

Appendix D

DATA POINTS LIST

DATA SYSTEM
INPUT LISTING

D(#)	UNIT/SLOT Counter	HOUSE CABLE	DESCRIPTION
1	0/0		BINo (Heat pump-to-bin heat flow)
2	0/1		BINi (Bin-to-heat pump heat flow)
3	0/2		SLPo (Solar panel heat rejection)
4	0/3		SLPi (Solar panel heat collection)
5	0/4	8	PCLG WHM Kh=3 (Cooling pump)
6	0/5	9	PHTG WHM Kh=3 (Heating pump)
7	0/6	10	PHWT WHM Kh=3 (Hot water pump)
8	0/7	6	COMP WHM Kh=3 (Compressor)
9	0/8	7	BLWR WHM Kh=3 (Blower)
10	0/9	16	ACES WHM Kh=7.2 (ACES total)
11	0/10		FANo (heat flow out of fan coil)
12	0/11		FANi (heat flow into fan coil)
13	0/12		HWTo (heat flow into hot water)
14	0/13		ECSD (Econ. cycle operating time, DH)
15	0/14		
16	1/0		ECSH (Econ. cycle operating time, H_{27})
17	1/1		
18	1/2		
19	1/3		SAMP (Solar air handler)
20	1/4		HPID (Heat pump indoor)
21	1/5		HPØD (Heat pump outdoor)
22	1/6		OFFP (Off peak heater)
23	1/7		HPAX (Heat pump auxiliary heater)
24	1/8		SCSP (Heat flow-Solar collector)
25	1/9		HPSP (Heat pump - Space)
26	1/10		SCPB (Solar collector pebble bed)
27	1/11		PBSP (Pebble Bed - Space)
28	1/12		TPWR (Total power)
29	1/13		
30	1/14		Digital Input
31	2/0		SOLC (solar collector heat flow)
32	2/1		WTRC (water-to-refrigerant coil heat flow)
33	2/2		WTAC (water-to-air coil heat flow)
34	2/3		HWTR (hot water heat flow)
35	2/4	116	SOLR WHM Kh= ___ (solar total)
36	2/5	103	BLWR WHM Kh=3 (Blower)
37	2/6	101	SHWT WHM Kh=3 (hot water heater)
38	2/7	104	PSCL WHM Kh=3 (collector pump)
39	2/8	105	DHTR WHM Kh=12 (duct heater)
40	2/9	102	PHTG WHM Kh=3 (heating pump)
41	2/10	115	HTPU WHM Kh=___ (heat pump outdoor unit)
42	2/11		RTAC (Refrigerant-to-air coil heat flow)

D(#)	UNIT/SLOT Counter	HOUSE CABLE	DESCRIPTION
43	2/12		
44	2/13		
45	2/14		
46	3/0		QRAD (Solar radiation, 45°)
47	3/1		WSPD (Average windspeed)
48	3/2		CHTP WHM Kh=7.2 (Control house total)
49	3/3		CHHW WHM Kh=3.6 (hot water heater
50	3/4		CHIU WHM Kh=12 (indoor unit)
51	3/5		CHOU WHM Kh=3.6 (outdoor unit)
52	3/6		FANi (heat flow into fan coil)
53	3/7		FANo (heat flow out of fan coil)
54	3/8		
55	3/9		
56	3/10		
57	3/11		
58	3/12		
59	3/13		
60	3/14		

D(#)	ANALOG Scanner	HOUSE CABLE	DESCRIPTION
61	0	14	Ice Level
62	1	23	BINl (Bin Lo Temp.)
63	2	22	BINh (Bin High Temp.)
64	3	3	BINi (Bin Inlet Temp.)
65	4	4	BINo (Bin Outlet Temp.)
66	5	51	FANi (Fan Coil Inlet Temp.)
67	6	52	FANo (Fan Coil Outlet Temp.)
68	7	55	SLPi (S.P. Inlet Temp.)
69	8	2	SLPo (S.P. Outlet Temp.)
70	9	53	HWTi (HW Inlet Temp.)
71	10	54	HWTo (HW Outlet Temp.)
72	11	(45)	H dry (House DB Temp.)
73	12	43	H wet (House WB Temp.)
74	13	42	SLPu (Solar Panel Unpumped Temp.)
75	14		
76	15		
77	16		
78	17		
79	18		
80	19		PBTl (Pebble bed #1 Temp.)
81	20		PBT2 (Pebble bed #2 temp.)
82	21		PBT3 (Pebble bed #3 temp.)
83	22		PBT4 (Pebble bed #4 temp.)
84	23		PBT5 (Pebble bed #5 temp.)
85	24		PBT6 (Pebble bed #6 temp.)
86	25		PBT7 (Pebble bed #7 temp.)
87	26		HDRY (Garage dry bulb temp.)
88	27		HWET (Garage wet bulb temp.)

D(#)	ANALOG Scanner		HOUSE CABLE	DESCRIPTION
89	28			
90	29			
91	30			
92	31			
93	32			
94	33			
95	34			
96	35	Red, Yel, Blk	310	Rain (Rainfall)
97	36	Blk, Yel, Grn	310	Wdir (Wind Direction, 0° ≡ N)
98	37	Blk, Org, Blu	310	Pdar (Barometric Pressure)
99	38	Red, Wht, Grn	310	Tdew (Outdoor dew point)
100	39	Blk, Grn, Wht	310	Tdry (Outdoor Dry Bulb)
101	40			
102	41		111	SLPi (Collector Inlet Temp.)
103	42		112	SLPo (Collector Outlet Temp.)
104	43		131	HWT_t (Hot water tank temp., top)
105	44		132	SLPu (Solar panel unpumped temp.)
106	45		133	SLPp (Solar panel pumped temp.)
107	46		134	HWTb (Hot water tank temp., bottom)
108	47		106	T#2h (Tank #2 High Temp.)
109	48		107	T#2L (Tank #2 Low Temp.)
110	49		108	T#1h (Tank #1 High Temp.)
111	50		109	T#1L (Tank #1 Low Temp.)
112	51		117	FANi (Fan coil inlet temp.)
113	52		118	FANo (Fan coil outlet temp.)
114	53		110	Hdry (House dry bulb temp.)
115	54		113	Hwet (House Wet bulb temp.)
116	55			
117	56			
118	57			
119	58			
120	59			
121	60			
122	61			
123	62			
124	63			
125	64			
126	65			
127	66			
128	67			
129	68			
130	69			
131	70			
132	71			
133	72			
134	73			
135	74			AIRo (Air temp. leaving indoor unit)
136	75			FANi (Fan coil inlet temp.)
137	76			FANo (Fan coil outlet temp.)
138	77		250	Hdry (House dry bulb temp.)

D(#)	ANALOG	HOUSE CABLE	DESCRIPTION
139	78	247	Hwet (House wet bulb temp.)
140	79	245	CRLS (Crawl space temp.)
141			
142			
143			
144			
145			
146			
147			
148			
149			
150			
151			
152			
153			
154			
155			
156			
157			
158			Stime
159			Month Day
160			Hour Min

Appendix E

PROCESSING A SAMPLE WEEK

Appendix E.1

CASSETTE FILE

Cassette file stored on DEC tape is "C80077.DAT."
167 hours in total week
begins 3/17/1980 1200. hours
ends 3/24/1980 1000. hours

SPDA data set is

"T.IJL05936.C80077"

Listing of output from program PUNCH is on microfiche in the pocket on the inside back cover.

Appendix E.2

DISK FILE

Filed data on SPDA is

"T.IJL05936.D80077"

on UTACE2 (private 2314 disk)

"ACES.DATA.D80077"

Listing of file processed by FILER and copied by CØPY.

Summarized results output to printer for weekly notebook.

(Display pages from SUMMARY and SØLAR.)

***** S U M M A R Y O F F I L E F O R 3/17/80 *****

*** LISTING OF HOURS MISSING FOR INDIVIDUAL RECORDS ***
(DOES NOT INCLUDE THOSE LISTED PREVIOUSLY)

--- THERE ARE NO ADDITIONAL HOURS MISSING ----

***** S U M M A R Y O F F I L E F O R 3/17/80 *****

*** LISTING OF HOURS FOR WHICH THERE IS NO DATA ***

--- THERE ARE NO MISSING HOURS ----

***** A N N U A L C Y C L E E N E R G Y S Y S T E M D E M O N S T R A T I O N P R O J E C T *****
ANALYZED FOR WEEK BEGINNING - 03/17/80

***** E Q U I P M E N T A N D W E A T H E R D A T A S U M M A R Y *****
(ID DATE NUMBERS ARE IN PARENTHESES)

AVERAGE OUTSIDE DRY BULB TEMPERATURE (C) ----------	8.55 (	8.55)
MAXIMUM OUTSIDE DRY BULB TEMPERATURE (C) ----------	18.35 (	18.35)
MINIMUM OUTSIDE DRY BULB TEMPERATURE (C) ----------	-2.09 (	-2.09)
DEGREE DAYS HEATING REFERENCED TO 18 (C) ----------	67.47 (	67.47)
DEGREE DAYS HEATING REFERENCED TO 15 (C) ----------	53.47 (	53.47)
DEGREE DAYS HEATING REFERENCED TO 14 (C) ----------	39.47 (	39.47)
DEGREE HOURS COOLING REFERENCED TO 21 (C) ----------	0.00 (	0.00)
DEGREE HOURS COOLING REFERENCED TO 23 (C) ----------	0.00 (	0.00)
DEGREE HOURS COOLING REFERENCED TO 25 (C) ----------	0.00 (	0.00)
TOTAL HEATING LOAD DELIVERED (KWH) ----------	344.27 (	344.27)
TOTAL COOLING LOAD DELIVERED (KWH) ----------	0.00 (	0.00)
TOTAL ECONOMY LOAD DELIVERED (KWH) ----------	0.00 (	0.00)
TOTAL HOT WATER LOAD DELIVERED (KWH) ----------	111.14 (	111.14)
MAXIMUM HEATING LOAD DELIVERED (WATT) ----------	6363.39 (	6363.39)
MAXIMUM COOLING LOAD DELIVERED (WATT) ----------	0.00 (	0.00)
MAXIMUM ECONOMY LOAD DELIVERED (WATT) ----------	0.00 (	0.00)
MAXIMUM HOT WATER LOAD DELIVERED (WATT) ----------	9365.00 (	9365.00)
TOTAL ACES POWER CONSUMPTION (KWH) ----------	176.58 (	176.58)
TOTAL CONTROL HOUSE EQUIP POWER CONSUMPTION (KWH) --	267.18 (	267.18)

***** A N N U A L C Y C L E E N E R G Y S Y S T E M D E M O N S T R A T I O N P R O J E C T *****
ANALYZED FOR WEEK BEGINNING - 03/17/80

**** L E A S T S Q U A R E D F I T F O R C O O L I N G L O A D S *****

SOLAR < 50 (BTU/HR-SQFT): THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION

SOLAR < 150 (BTU/HR-SQFT): THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION

SOLAR > 150 (BTU/HR-SQFT): THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION

NO SOLAR DIFFERENTIATION: THERE IS INSUFFICIENT DATA TO CALCULATE THE EQUATION

**** L E A S T S Q U A R E D F I T F O R H E A T I N G L O A D S *****

SOLAR < 50 (BTU/HR-SQFT): Q (WATTS) = -2.19566E+02 * TEMPERATURE(C) + 4.28431E+03
BREAK EVEN TEMPERATURE = 19.5125E+00 (C) NO. OF POINTS ANALIZED = 83

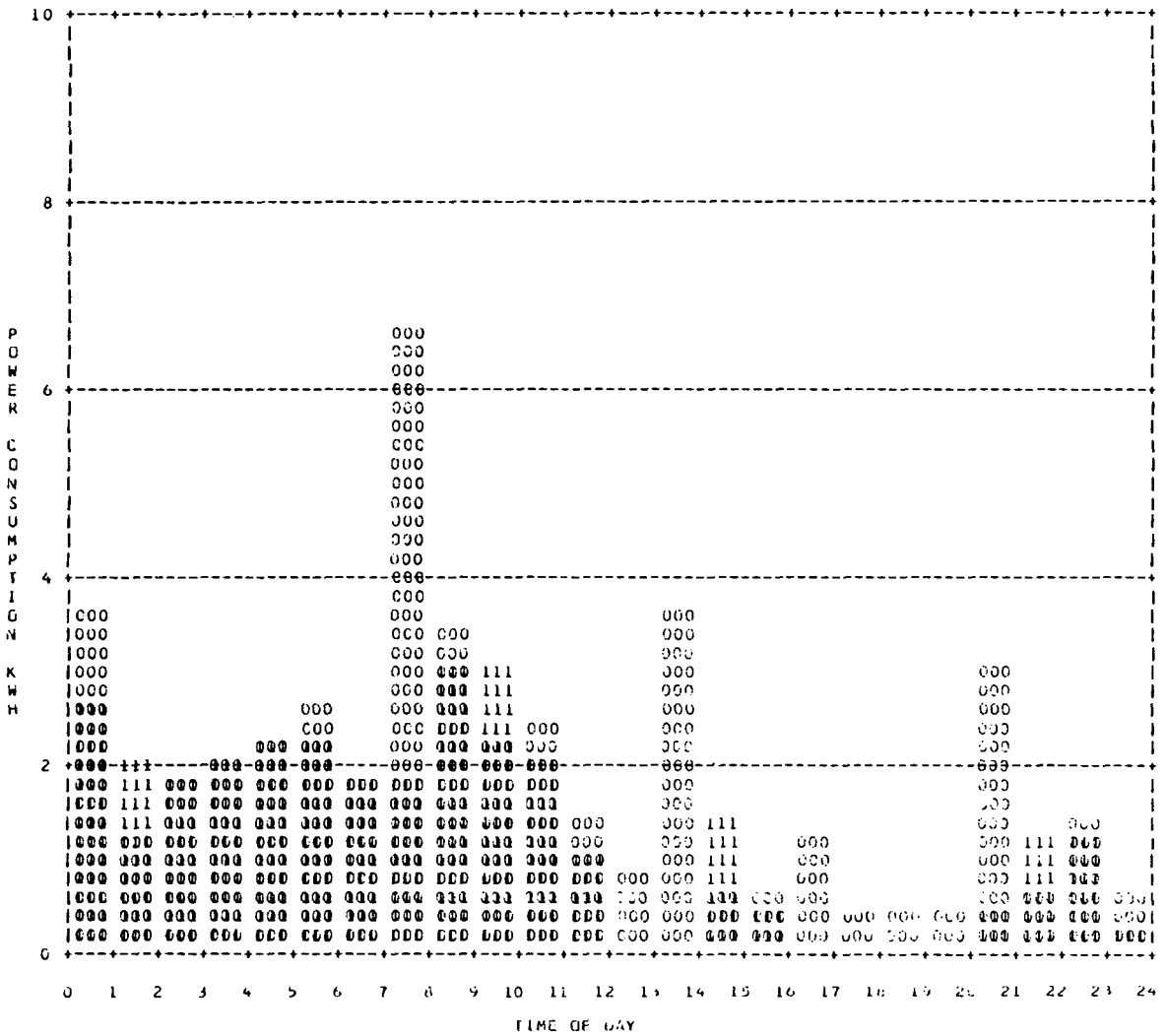
SOLAR < 150 (BTU/HR-SQFT): Q (WATTS) = -3.89785E+02 * TEMPERATURE(C) + 6.41326E+03
BREAK EVEN TEMPERATURE = 16.4456E+00 (C) NO. OF POINTS ANALIZED = 13

SOLAR > 150 (BTU/HR-SQFT): Q (WATTS) = 8.05114E+00 * TEMPERATURE(C) + 1.63542E+03
BREAK EVEN TEMPERATURE = -20.312?E+01 (C) NO. OF POINTS ANALIZED = 30

NO SOLAR DIFFERENTIATION: Q (WATTS) = -2.35912E+02 * TEMPERATURE(C) + 4.45343E+03
BREAK EVEN TEMPERATURE = 18.8777E+00 (C) NO. OF POINTS ANALIZED = 126

***** ANNUAL CYCLE ENERGY SYSTEM DEMONSTRATION PROJECT *****
ANALYZED FOR WEEK BEGINNING - 03/17/80

WORST DAY POWER USE (0=CONTROL HOUSE, 1=ACES HOUSE)



***** SOLAR ENERGY SYSTEM DEMONSTRATION PROJECT *****
ANALYZED FOR WEEK BEGINNING - 3/17/80

***** EQUIPMENT AND WEATHER DATA SUMMARY *****

TOTAL SOLAR ENERGY COLLECTED	0.0	KWH
SOLAR SPACE HEATING ENERGY DELIVERED	0.00	KWH
ELECTRIC SPACE HEATING ENERGY DELIVERED	0.0	KWH
TOTAL SPACE HEATING ENERGY DELIVERED	0.00	KWH
SOLAR WATER HEATING ENERGY DELIVERED	11.95	KWH
ELECTRIC WATER HEATING ENERGY DELIVERED	0.0	KWH
TOTAL WATER HEATING ENERGY DELIVERED	0.0	KWH
TOTAL SOLAR SYSTEM ELECTRICITY	0.0	KWH
TOTAL HEATING AND HOT WATER ELECTRICITY CONSUMPTION	11.95	KWH
MINIMUM INDOOR TEMPERATURE FOR SOLAR HOUSE	12.95	C

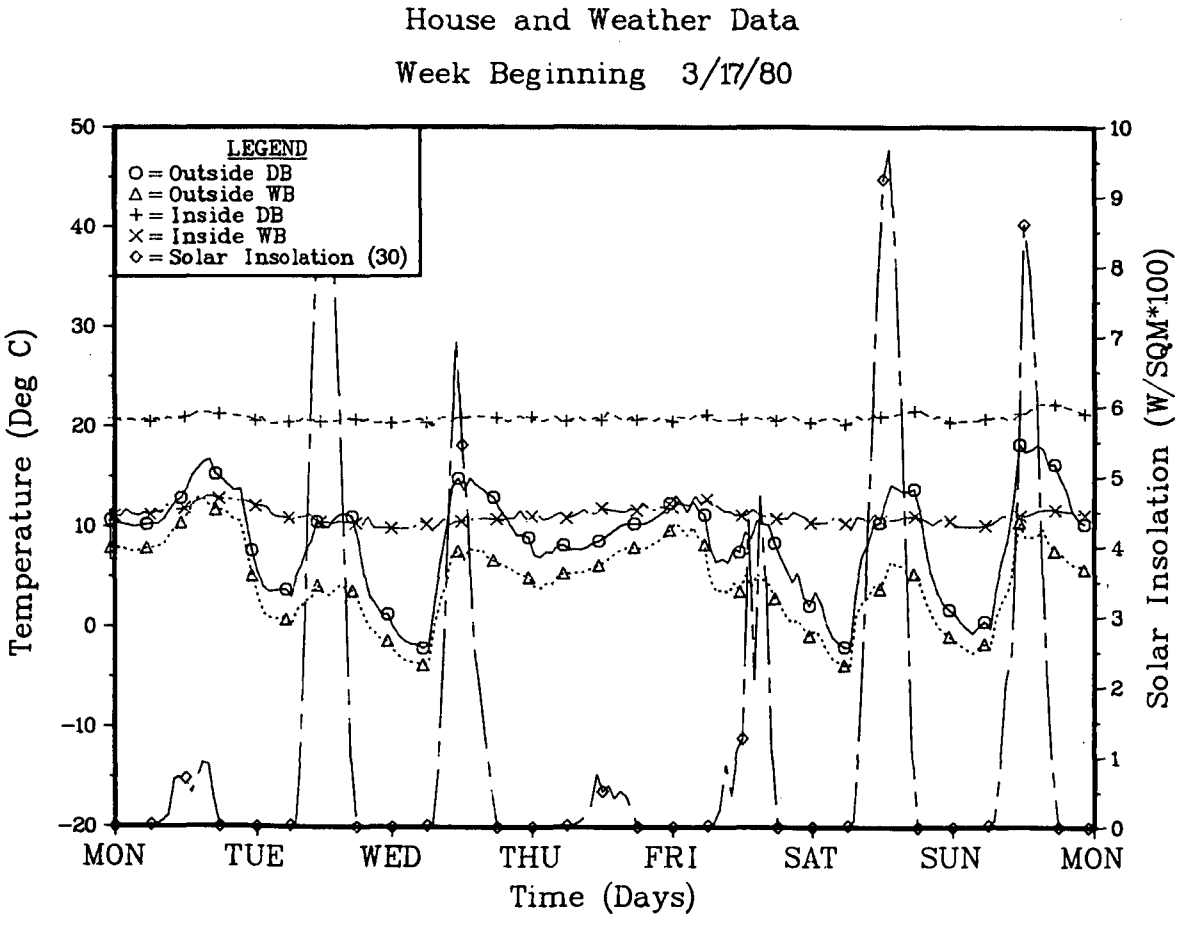
IHC002I STOP 0

Appendix E.3

OUTPUT FROM PROGRAM PLOT

Plotted curves result in output from PLOT, filed in the weekly notebook. The following nine plots are illustrative of this output.

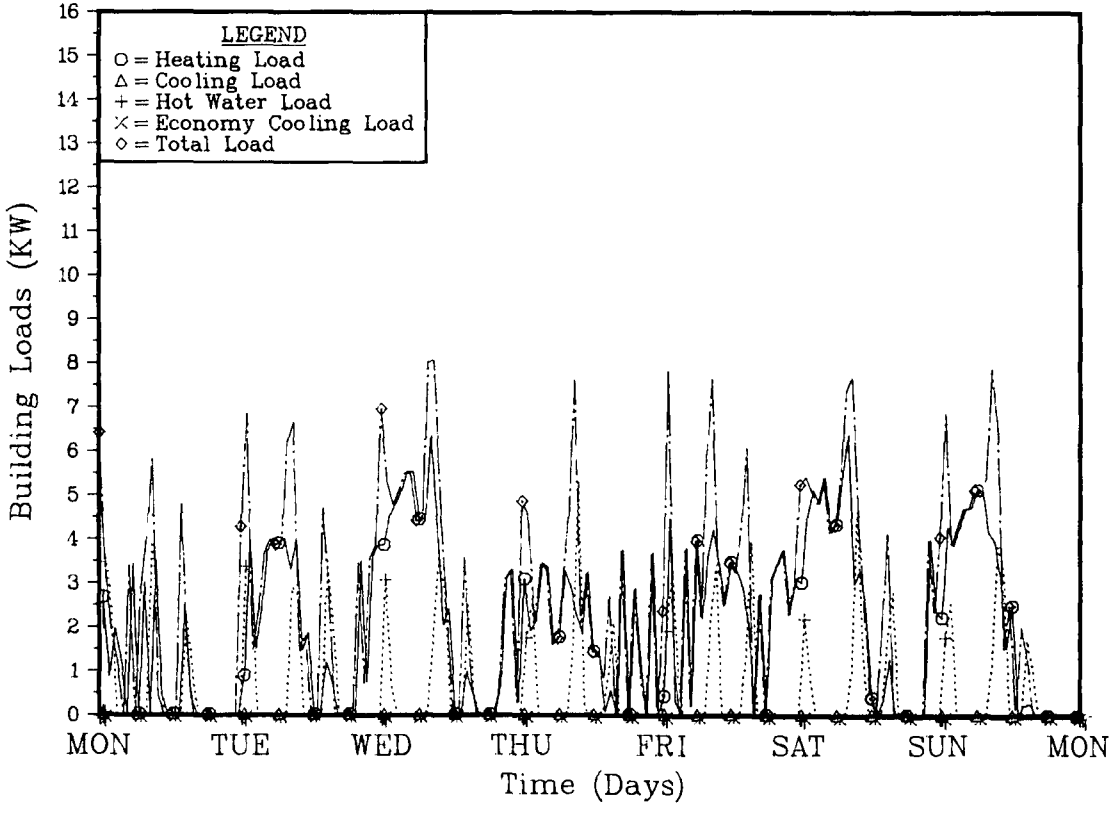
ORNL–DWG 81-6514



ORNL–DWG 81-6513

ACES Loads

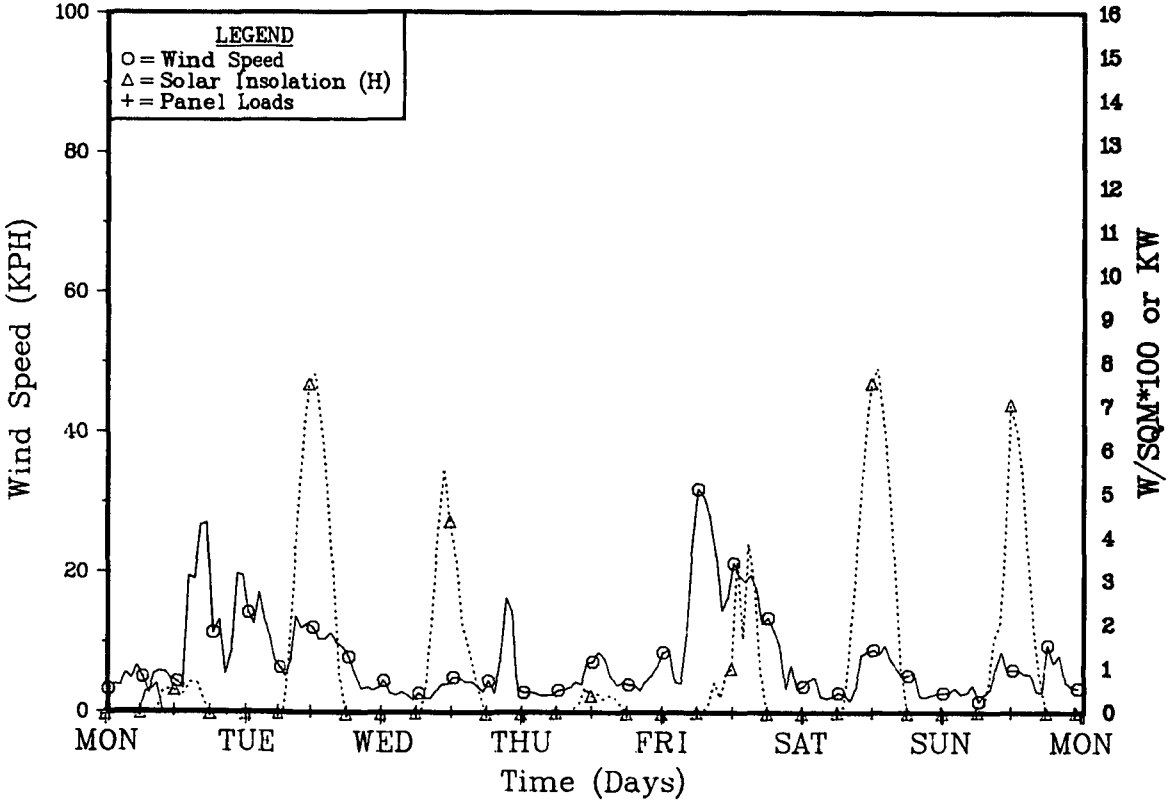
Week Beginning 3/17/80



ORNL–DWG 81-6512

Solar Panel Analysis

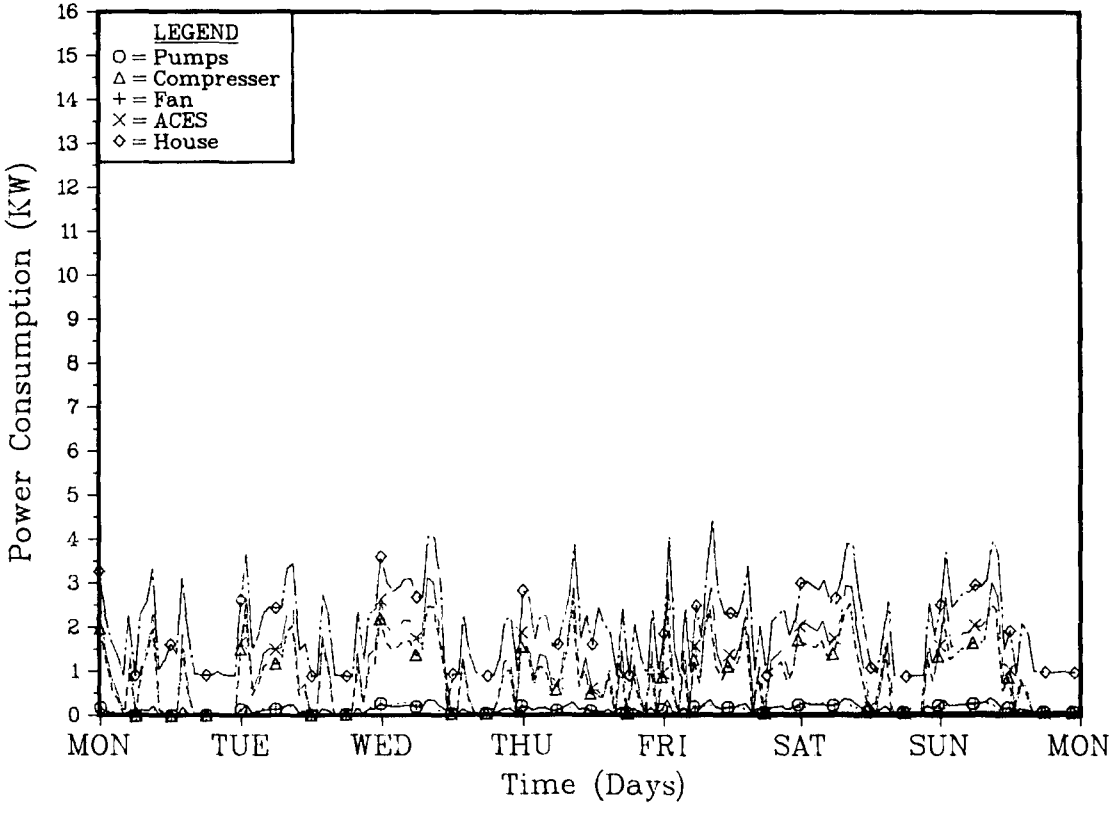
Week Beginning 3/17/80



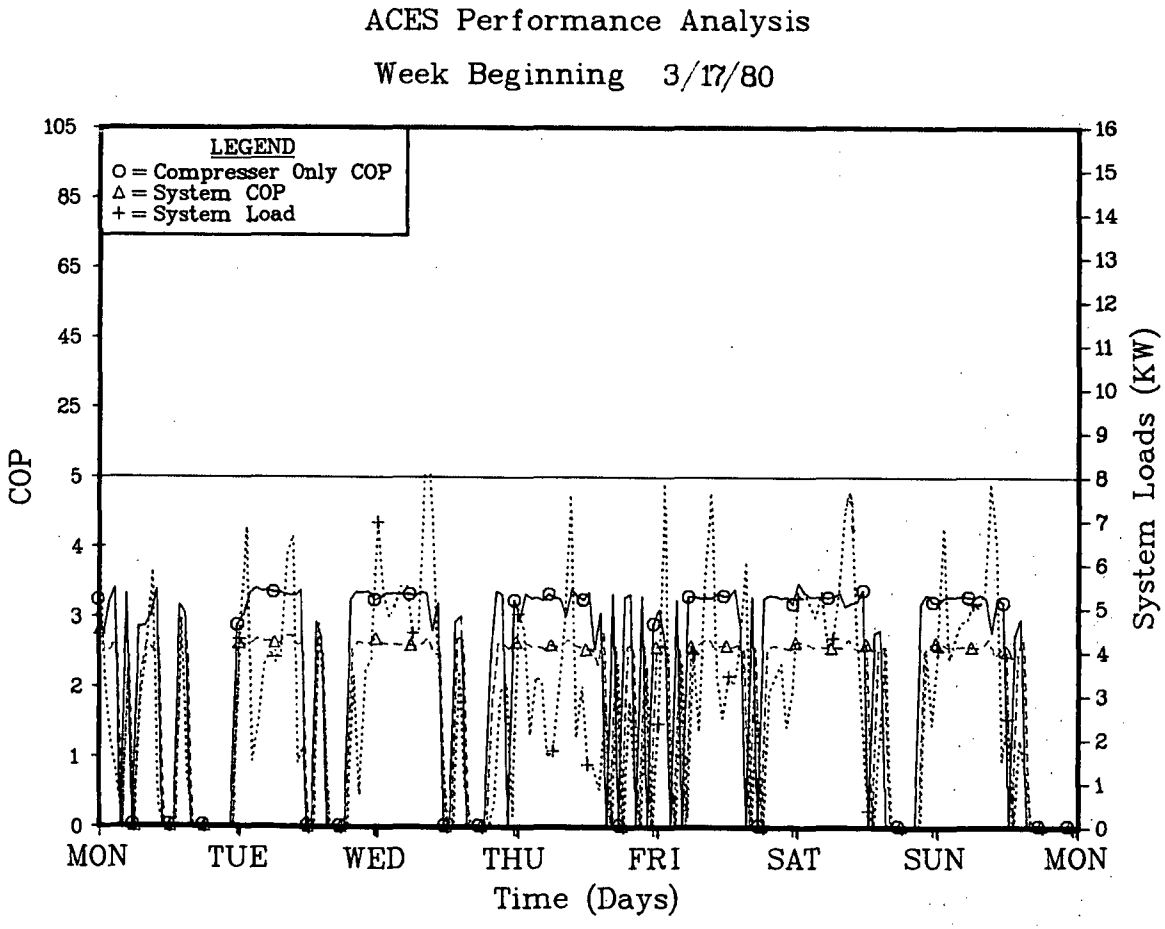
ORNL–DWG 81-6518

ACES Electrical Analysis

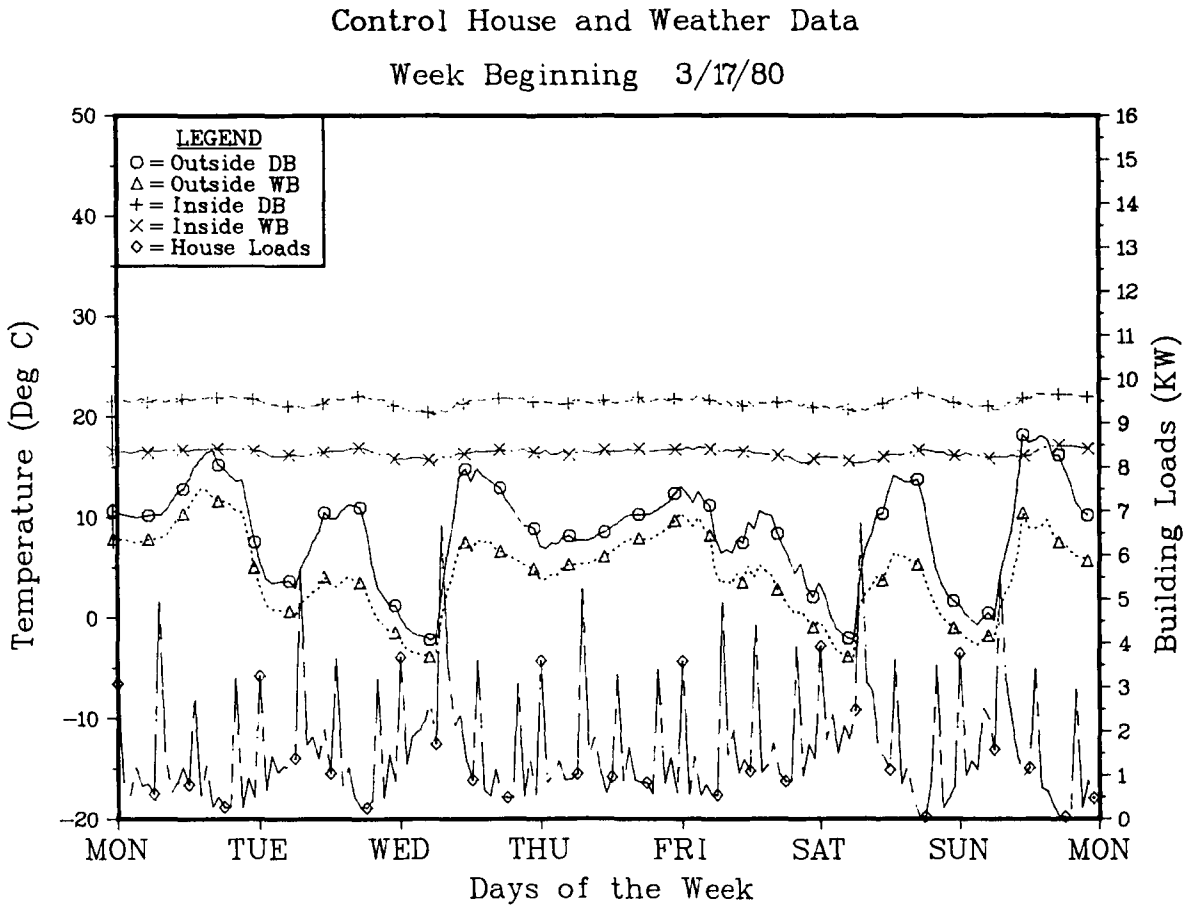
Week Beginning 3/17/80



ORNL–DWG 81-6511



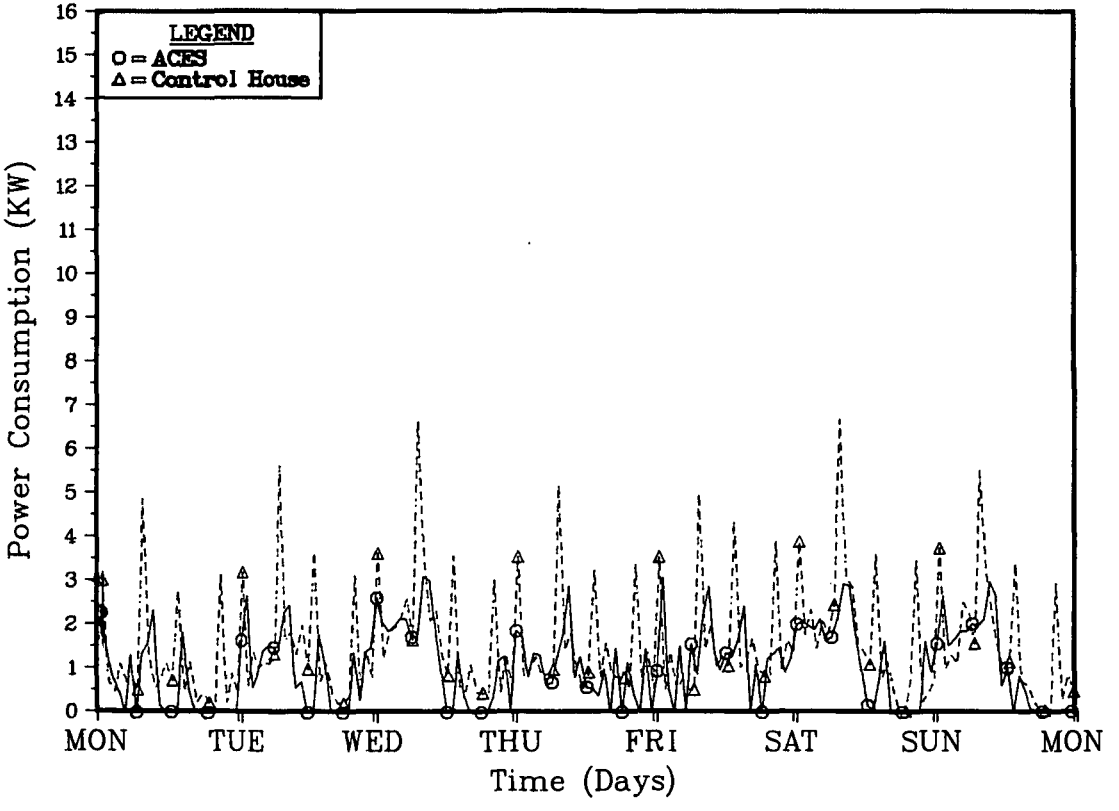
ORNL–DWG 81-6510



ORNL–DWG 81-6515

Comparison of ACES and Control House

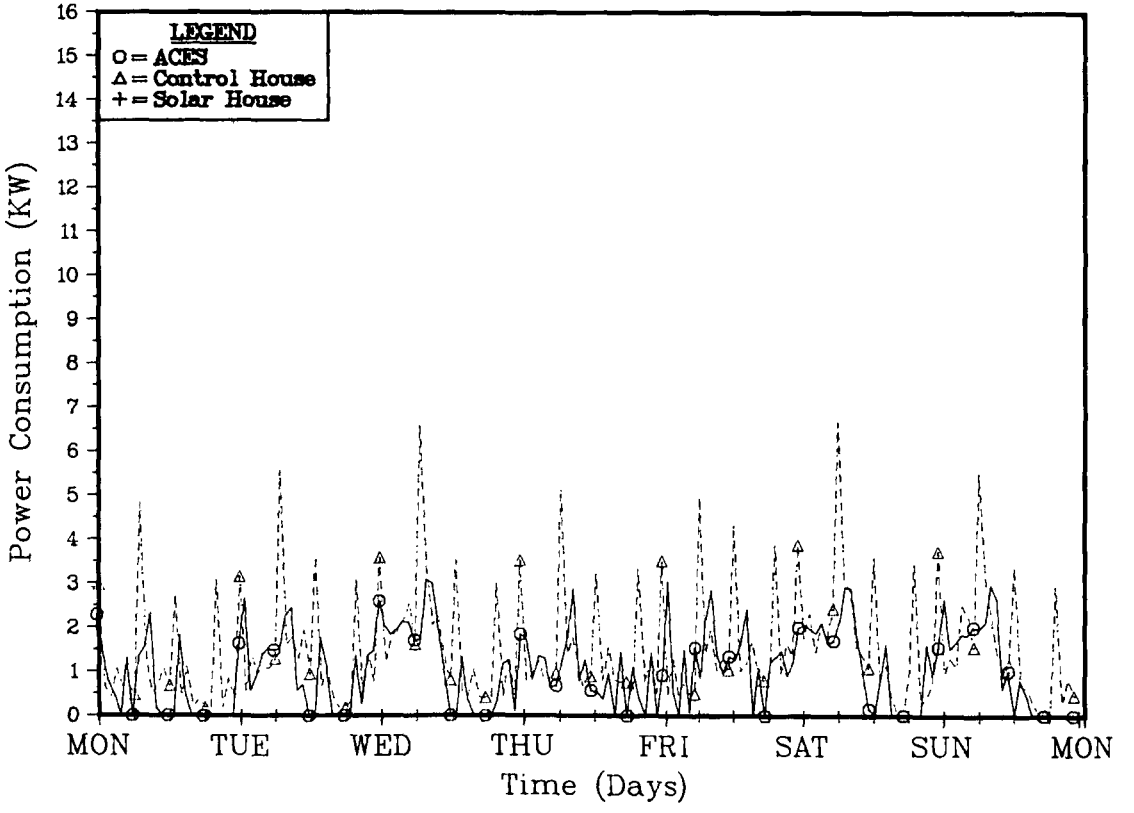
Week Beginning 3/17/80



ORNL–DWG 81-6516

Comp of ACES, Solar, and Control House

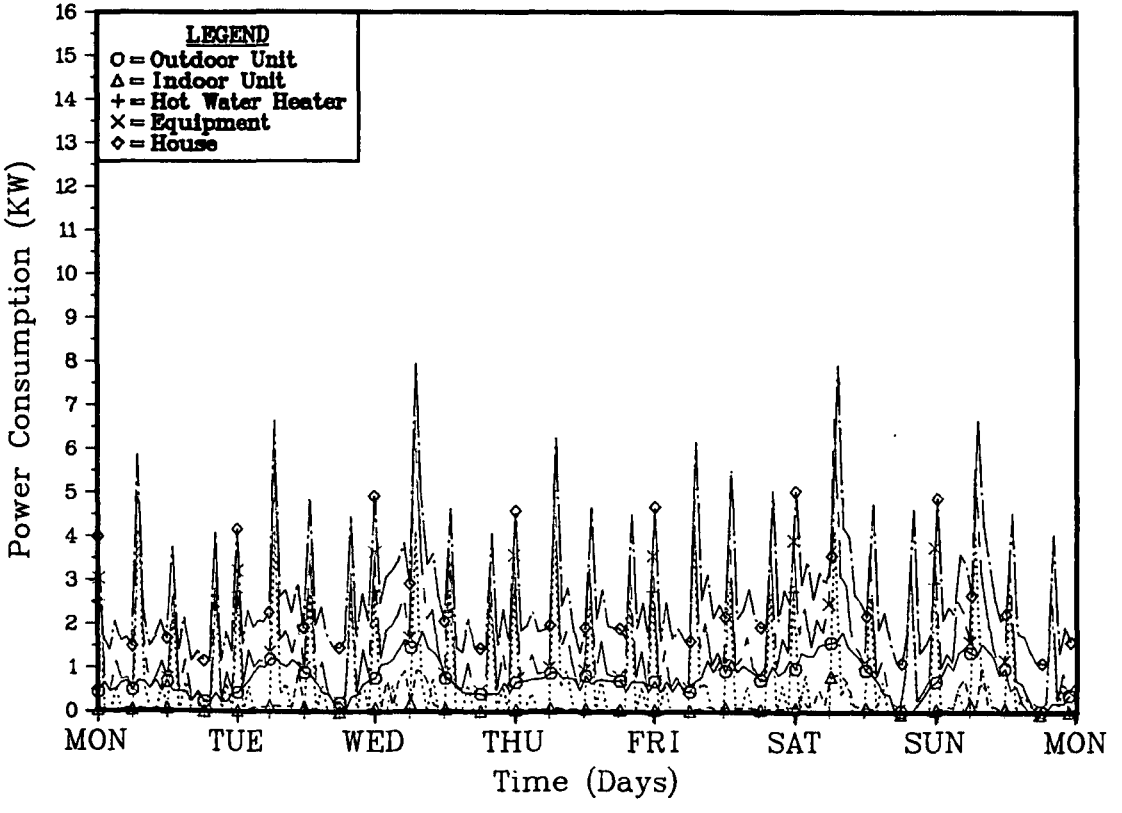
Week Beginning 3/17/80



ORNL–DWG 81-6517

Control House Electrical Analysis

Week Beginning 3/17/80



Appendix E.4

MONTHLY UPDATE

The following output is from the monthly update program, MONTHLY.

```
EX CMDPROC(MONTHLY)
ENTER THE FIRST MONTH TO BE PRINTED
01
ENTER THE LAST MONTH TO BE PRINTED
12
DO YOU WANT AN ECONOMY CYCLE
   (0=NO/1=TEMPERATURE CONTROL/2=HUMIDITY CONTROL)
0
```

JANUARY

TOTAL MONTHLY HEATING LOAD (BTU)	10225261.00
TOTAL MONTHLY COOLING LOAD (BTU)	19317.35
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1673790.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	15460.66
TOTAL MONTHLY DEGREE DAYS HEATING	1032.66
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	31.69
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	744.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	744.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	592.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	476.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	149.00
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	11.00

FEBRUARY

TOTAL MONTHLY HEATING LOAD (BTU)	7224975.00
TOTAL MONTHLY COOLING LOAD (BTU)	51570.99
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1642705.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	12949.07
TOTAL MONTHLY DEGREE DAYS HEATING	792.73
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	36.60
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	672.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	670.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	434.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	352.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	86.00
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	11.00

MARCH

TOTAL MONTHLY HEATING LOAD (BTU)	3240331.00
TOTAL MONTHLY COOLING LOAD (BTU)	43797.38
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1654694.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	24478.00
TOTAL MONTHLY DEGREE DAYS HEATING	369.16
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	50.82
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	744.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	625.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	155.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	83.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	1.00
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

APRIL

TOTAL MONTHLY HEATING LOAD (BTU)	761601.88
TOTAL MONTHLY COOLING LOAD (BTU)	19955.43
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1391202.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	33767.72
TOTAL MONTHLY DEGREE DAYS HEATING	161.45
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	57.88
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	714.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	544.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	21.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	6.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

MAY

TOTAL MONTHLY HEATING LOAD (BTU)	268675.69
TOTAL MONTHLY COOLING LOAD (BTU)	878054.69
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1664718.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	38596.25
TOTAL MONTHLY DEGREE DAYS HEATING	0.0
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	65.15
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	706.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	361.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

JUNE

TOTAL MONTHLY HEATING LOAD (BTU)	136126.13
TOTAL MONTHLY COOLING LOAD (BTU)	4265144.00
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1220829.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	45415.81
TOTAL MONTHLY DEGREE DAYS HEATING	0.0
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	71.09
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	587.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	182.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

JULY

TOTAL MONTHLY HEATING LOAD (BTU)	98367.44
TOTAL MONTHLY COOLING LOAD (BTU)	4835002.00
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1185516.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	35530.11
TOTAL MONTHLY DEGREE DAYS HEATING	0.0
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	73.14
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	604.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	48.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

AUGUST

TOTAL MONTHLY HEATING LOAD (BTU)	62115.48
TOTAL MONTHLY COOLING LOAD (BTU)	6802165.00
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1316450.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	41636.25
TOTAL MONTHLY DEGREE DAYS HEATING	0.0
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	74.57
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	524.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	78.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

SEPTEMBR

TOTAL MONTHLY HEATING LOAD (BTU)	194103.56
TOTAL MONTHLY COOLING LOAD (BTU)	4153948.00
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1600065.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	26491.77
TOTAL MONTHLY DEGREE DAYS HEATING	0.0
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	69.12
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	651.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	236.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

OCTOBER

TOTAL MONTHLY HEATING LOAD (BTU)	451939.63
TOTAL MONTHLY COOLING LOAD (BTU)	143337.94
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	896748.69
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	25624.23
TOTAL MONTHLY DEGREE DAYS HEATING	190.31
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	56.93
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	735.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	566.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	45.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	19.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	0.0
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

NOVEMBER

TOTAL MONTHLY HEATING LOAD (BTU)	2318360.00
TOTAL MONTHLY COOLING LOAD (BTU)	49944.64
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1373080.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	14745.74
TOTAL MONTHLY DEGREE DAYS HEATING	442.09
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	49.14
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	720.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	669.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	150.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	103.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	9.00
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

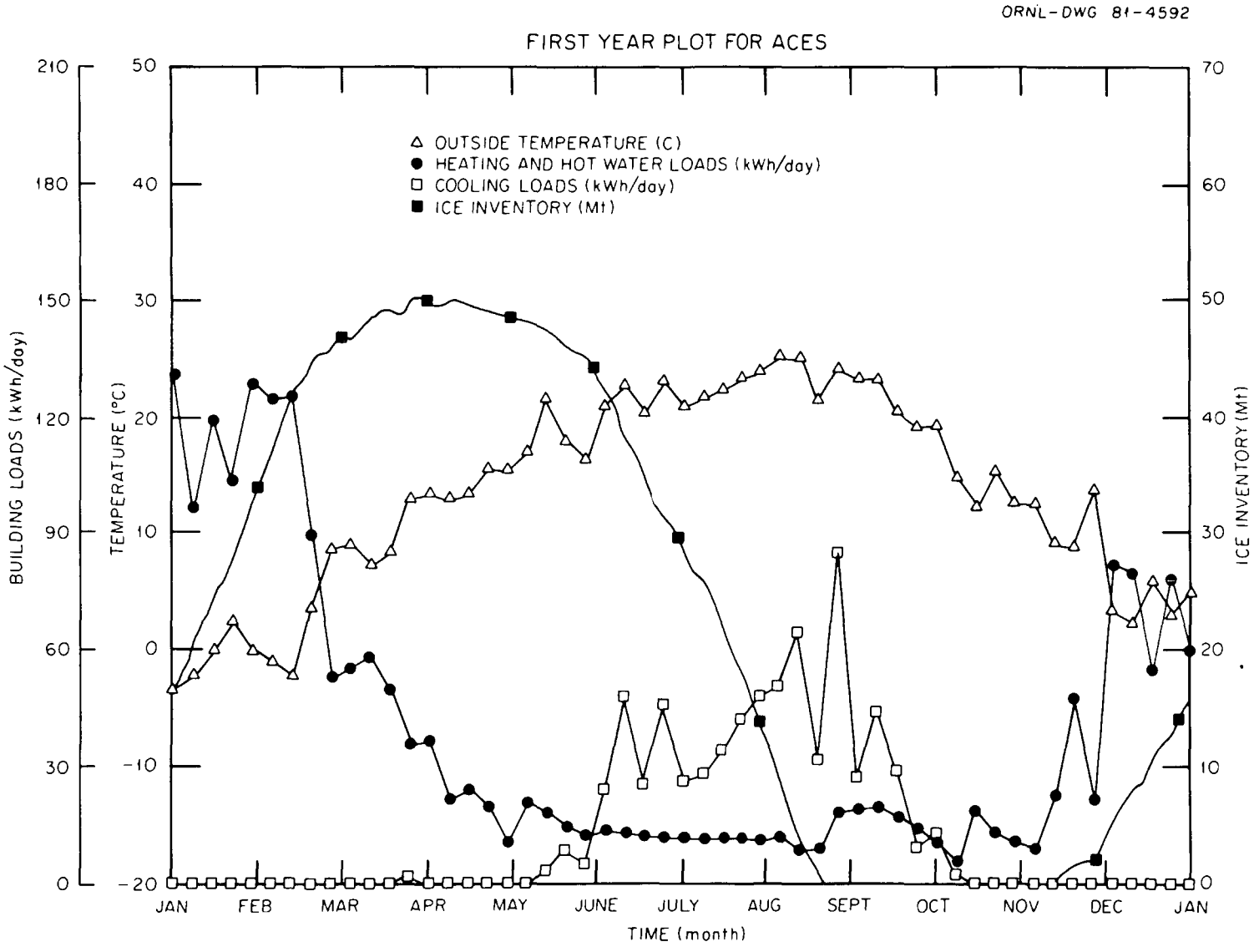
DECEMBER

TOTAL MONTHLY HEATING LOAD (BTU)	6486416.00
TOTAL MONTHLY COOLING LOAD (BTU)	4113.51
TOTAL MONTHLY DOMESTIC HOT WATER LOAD (BTU)	1557522.00
TOTAL MONTHLY SOLAR RADIATION (BTU/FT/FT)	13518.64
TOTAL MONTHLY DEGREE DAYS HEATING	816.74
AVERAGE MONTHLY DRY BULB TEMPERATURE (F)	38.51
TOTAL MONTHLY HOURS LESS THAN 80 DEG F	744.00
TOTAL MONTHLY HOURS LESS THAN 65 DEG F	740.00
TOTAL MONTHLY HOURS LESS THAN 39 DEG F	396.00
TOTAL MONTHLY HOURS LESS THAN 35 DEG F	278.00
TOTAL MONTHLY HOURS LESS THAN 25 DEG F	62.00
TOTAL MONTHLY HOURS LESS THAN 10 DEG F	0.0

Appendix E.5

ICE INVENTORY PLOT

The following illustration shows the first-year plot for the ACES ice inventory.


ORNL-DWG 81-4592
FIRST YEAR PLOT FOR ACES
OUTSIDE TEMPERATURE (C)
HEATING AND HOT WATER LOADS (kWh/day)
COOLING LOADS (kWh/day)
ICE INVENTORY (Mt)
BUILDING LOADS (kWh/day)
TEMPERATURE (°C)
ICE INVENTORY (Mt)
210
180
150
120
90
60
30
0
50
40
30
20
10
0
-10
-20
70
60
50
40
30
20
10
0
JAN
FEB
MAR
APR
MAY
JUNE
JULY
AUG
SEPT
OCT
NOV
DEC
JAN
TIME (month)

Internal Distribution

1. L. A. Abbatiello
2. R. K. Adams
3. V. D. Baxter
4. J. L. Bledsoe
5. A. A. Brooks
6. N. S. Cardell
7. R. S. Carlsmith
8. H. P. Carter
9. B. A. Clark
10. G. E. Courville
11. J. S. Crowell
12. J. W. Dick
13. R. D. Ellison
14. S. K. Fischer
15. W. Fulkerson
16. E. L. Hillsman
17. W. L. Jackson
18. C. G. Lawson
19. W. P. Levins
20. A. A. Loebl
21. J. F. Martin
22. B. W. McConnell
23. J. A. McEvers
24. G. S. McNeilly
25. J. W. Michel
26. D. R. Miller
27. B. D. Murphy
28. E. A. Nephew
29. D. O'Neal
30. S. C. Parikh
31. M. R. Patterson
32. D. T. Rizy
33. R. C. Robertson
34. C. A. Sady
35. W. L. Zabriskie
36. Biology Division Library
37-38. Central Research Library
39. Document Reference Section
40-41. Energy Information Library
42. Laboratory Records - RC
43-44. Laboratory Records Department
45. ORNL Patent Section

External Distribution

46. Roy W. Abbott, Manager, Advanced Technology, General Electric Company, Appliance Park, Building 6-208, Louisville, KY 40225
47. Ben Adams, Adams Craft Herz Walker, 106 Administration Road, Oak Ridge, TN 37830
48. Kent Anderson, Trane Company, 2020 14th Street, Arlington, VA 22201
49. Bruce Appelbaum, Gordian Associates Incorporated, 711 Third Avenue, New York, NY 10017
50. Arthur G. Baitz, Assistant Vice President and Director of Planning, Robertshaw Controls Company, 1701 Byrd Avenue, P.O. Box 26544, Richmond, VA 23261
51. Hamp Barnett, Power Research Staff, Tennessee Valley Authority, 350 Commerce Union Bank Building, Chattanooga, TN 37401
52. Larry B. Barrett, Potomac Electric Power Company, Manager, Energy Management Services Division, 1900 Pennsylvania Avenue NW, Washington, DC 20068
53. A. F. G. Bedinger, Energy, Environment, and Resources Center, University of Tennessee, 329 South Stadium Building, Knoxville, TN 37916
54. David A. Bell, Research Engineer, Utah State University, Department of Mechanical Engineering, UCM-41, Logan, UT 84322
55. John Blair, 1019 Bellemeade, Evansville, IN 47714
56. Thomas P. Bligh, Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN 55455

57. Gene Brewer, NIBS, Suite 700, 1015 15th Street, NW, Washington, DC 20005
58. William B. Briggs, McDonnell-Douglas Company, Department E080, P.O. Box 516, St. Louis, MO 63166
59. Thomas Cahill, Energy Systems Division, Carrier, Carrier Tower, P.O. Box 4800, Syracuse, NY 13221
60. Joe Canal, General Electric Company, Troup Highway, Tyler, TX 75711
61. Louis-Marie Chounet, Batiment 200, Lab de l'Accelerateur Lineaire, 91405 Orsay, France
62. W. James Cole, Program Director, New York State Energy Research and Development Authority, Agency Building No.2, Empire State Plaza, Albany, NY 12223
63. David Colvin, Triangle Research and Development Corporation, P.O. Box 12696, Research Triangle Park, NC 27709
64. Ronald Cosby, Department of Physics and Astronomy, Ball State University, Muncie, IN 47306
65. John J. Cuttica, Technology and Consumer Products Branch, MS GH-068, Department of Energy, 1000 Independence Avenue SW, Washington, DC 20585
66. H. E. Davis, Concentration Specialists, Inc., 26 Dundee Park, Andover, MA 01810
67. Gordon D. Duffy, Executive Editor, Air Conditioning News, 700 East Maple, Birmingham, MI 48011
68. William A. Elder, President, Automated Air Systems Inc., 1723 Rhoadmiller Street, Richmond, VA 23220
69. Harry C. Fischer, P.O. Box 1687, Cocoa, FL 32922
70. Douglas Funkhouser, Ruban Systems Research and Engineering Inc., 36 Boylston Street, Cambridge, MA 02138
71. Donald P. Gatley, Gatley and Associates, 489 Westover Drive NW, Atlanta, GA 30305
72. Ted Gillis, Lennox Industries, Promenade Towers, P.O. Box 400450, Dallas, TX 75240
73. George E. Gilmore, Consolidated Edison Co. of New York, Inc., 4 Irving Place, New York, NY 10003
74. L. R. Glicksman, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139
75. David Goldenberg, TRW Energy Systems Inc., 800 Oak Ridge Turnpike, Oak Ridge, TN 37830
76. Dr. R. Eugene Goodson, Director, Institute for Interdisciplinary Engineering Studies, Potter Engineering Center, Purdue University, West Lafayette, IN 47974
77. Dieter Grether, Vice President, Engineering, Friedrich Air Conditioning and Refrigeration, P.O. Box 1540, San Antonio, TS 78295
78. Gerald C. Groff, Research Division, Carrier Corporation, Syracuse, NY 13201
79. Donald G. Hoffard, P.O. Box 3221, 1002 N. E. Holladay, Portland, OR 97208
80. Kenneth C. Hoffman, Senior Vice President, Energy and Environment, Mathtech, 1611 North Kent Street, Arlington, VA 22209
81. F. Honnold, Consumer Products Division, Carrier Corporation, Syracuse, NY 13201

82. E. J. Horan, Exxon Research and Engineering Company, P.O. Box 8, Linden, NJ 07036
83. Günther Ivens, Arbeitsgemeinschaft Versuchsreaktor GmbH, 4000 Duesseldorf, Postfach 1411, Federal Republic of Germany
84. H. Jaster, General Electric Corporate Research and Development, GE Company, Schenectady, NY 12301
85. Ralph F. Jones, Brookhaven National Laboratory, Upton, Long Island, NY 11973
86. Ted Kapus, Director, Consumer Products Division, MS GH-068, Department of Energy, 1000 Independence Avenue SW, Washington, DC 20585
87. Richard M. Kelso, Associate Professor, School of Architecture, University of Tennessee, 100 Estabrook Hall, Knoxville, TN 37916
88. R. W. King, Manager, Advanced Engineering, Copeland Corporation, Sydney, OH 45365
89. William A. Knox, 111 Edgewood Avenue, Columbia, MO 65201
90. Sherman Kouns, Sales Representative, General Shale Products Corporation, P.O. Box 733, Knoxville, TN 37901
91. Edward A. Kush, Associate Scientist, Solar Research Group, Brookhaven National Laboratory, Upton, NY 11973
92. Dr. Todd R. LaPorte, Professor, Political Science, Institute of Government Studies, University of California at Berkeley, Berkeley, CA 94720
93. Jim Leggoe, Philadelphia Electric Company, 2301 Market Street, Philadelphia, PA 19101
94. Dan Lim, Program Manager, Technology and Consumer Products Branch, Department of Energy CS 113.2, Room GH-065, 1000 Independence Avenue, Washington, DC 20585
95. Herbert Lindahl, Manager, Advanced Engineering, Bohn Aluminum and Brass Corporation, Heat Transfer Division, 1625 East Voorhees Street, Danville, IL 61832
96. Mr. Quentin Looney, Energy Utilization and Conservation Technology, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
97. Harold G. Lorsch, Manager, Energy Laboratory Engineering Department, The Franklin Institute Research Laboratories, Twentieth and Parkway, Philadelphia, PA 19103
98. Hubert P. Luckett, Editor-in-Chief, *Popular Science,* 380 Madison Ave., New York, NY 10017
99. I. Ward MacArthur, Honeywell, Inc., 2600 Ridgeway Parkway, NE, Minneapolis, MN 55413
100. Merle McBride, Owens-Corning Technical Center, Route 16, P.O. Box 415, Granville, OH 43023
101. Jim McCallum, Executive Editor, Air Conditioning and Refrigeration Business, 614 Superior Avenue West, Cleveland, OH 44113
102. Jack McCormick, Research Engineer, Northern States Power Company, 414 Nicollet Mall, Minneapolis, MN 55401
103. Robert Mauro, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
104. J. P. Millhone, Director, Office of Buildings and Community Systems, CS-11, Department of Energy, 1000 Independence Avenue SW, Washington, DC 20585

105. Larry I. Moss, P.O. Box 1496, Estes Park, CO 80617
106. Dan Myers, Refrigeration Systems, 4241 Hogue Road, Evansville, IN 47712
107. Ken Nemeth, Executive Director, Southern States Energy Board, 2300 Peachford Road, One Exchange Place — Suite 1230, Atlanta, GA 30338
108. J. A. O'Brien, Director, Bedford Administrative Operation, The Mitre Corporation, Bedford, MA 01730
109. Hendrik van der Ree, TNO Heat Pump Section, Laan van Westenenk 501, Apeldoorn, The Netherlands
110. Bill Riley, Chief Engineer, Advance Development, Heil-Quaker Corporation, 647 Thompson Lane, P.O. Box 40566, Nashville, TN 37204
111. James K. Risher, Energy Resources Center, Honeywell, Inc., 2600 Ridgeway Parkway, Minneapolis, MN 55413
112. Dr. David J. Rose, Professor, Department of Nuclear Engineering, Room 24-210, Massachusetts Institute of Technology, Cambridge, MA 02139
113. A. H. Rosenfeld, Lawrence Berkeley Laboratory, Building 50B, Room 5246, Berkeley, CA 94720
114. David P. Ross, Senior Scientists, Energy Resources and Conservation Division, Science Applications, Inc., 2028 Powers Ferry Road, Suite 260, Atlanta, GA 30339
115. James L. Schulze, Technology Program Manager, General Electric Company Alliance Park, Louisville, KY 40225
116. Charles F. Sepsy, The Ohio State University Department of Mechanical Engineering, 206 W. 18th Avenue, Columbus, OH 43210
117. Charles Shattenkirk, Box 82H, RD2, Valatie, NY 12184
118. Dr. V. Kerry Smith, Professor, Department of Economics, University of North Carolina, Chapel Hill, NC 27514
119. Mason Somerville, Department of Mechanical Engineering, University of Arkansas, Fayetteville, AR 72701
120. E. Spannhake, White Consolidated Industries, 11770 Berea Road, Cleveland, OH 44111
121. Richard D. Stewart, Manager, Advanced Technology Programs, General Electric Company, Appliance Park (Bldg. 35) Louisville, KY 40225
122. Steven Strong, Solar Design Associates, 271 Washtington St., Canton, MA 02021
123. Floyd Stuckey, FSA Engineering Consultants, 315 East Sixth Street, Winfield, KS 67156
124. Walter D. Syniuta, Advanced Mechanical Technology, Inc., 141 California Street, Newton, MA 02158
125. Robert T. Tamblyn, Engineering Interface Limited, 1200 Sheppard Avenue East, Toronto, Ontario M2K2R8
126. S. S. Waddle, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830
127. Frank S. Wang, Research Specialist, Functional Products & Systems, Larkin Lab, Swede Road, Midland, MI 48640
128. H. F. Waser, FHP Manufacturing Company, 610 Southwest 12th Avenue, Pompano Beach, FL 33060
129. Robert Weatherwax, California Energy Commission, Mail Stop 23, 1111 Howe Avenue, Sacramento, CA 95825
130. Robert G. Werden, Werden & Associates, Inc., P.O. Box 414, Jenkintown, PA 19046
131. Frank White, AiResearch Manufacturing Company, P.O. Box 3413, Oak Ridge, TN 37830

132. M. Wilden, University of New Mexico, Albuquerque, NM 87115
133. Institute of Energy Analysis, ORAU Library
134. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830
135-161. Technical Information Center, Department of Energy, P.O. Box 62, Oak Ridge, TN 37830
162-564. External Energy Conservation Distribution Mailing List and Energy Conservation Office (9102-2, Conference Room)

☆U.S. GOVERNMENT PRINTING OFFICE: 1981-740-062/181

ORNL/CON-59

DATA COLLECTION AND PROCESSING FOR THE ACES

J. L. Bledsoe D. R. Miller