COMPUTER-AIDED CONCURRENT ENGINEERING IN REFRIGERATION SYSTEM DESIGN

by

Jeffrey L. Altenhof

Report submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

SYSTEMS ENGINEERING

APPROVED:

W.J. Fabrycky, Chairman

B.S. Blanchard

R.G. Mitchiner

April, 1990

Blacksburg, Virginia

COMPUTER-AIDED CONCURRENT ENGINEERING IN
REFRIGERATION SYSTEM DESIGN

by

Jeffrey L. Altenhof

Committee Chairman: Dr. W.J. Fabrycky
Industrial Engineering And Operations Research

(ABSTRACT)

Computer-Aided Concurrent Engineering Design (CACED) is an emerging field which stems from the realization that a holistic design approach, simultaneously considering all requirements, will result in systems that can be fielded quickly, and at the lowest practical lifetime cost. The philosophy inherent to CACED is that in a multi-faceted design arena, requirements such as cost, performance, reliability, produceability, size and supportability will conflict. Traditionally, designs are established then "audited" for compliance with various requirements. Subsequent "corrections" might then create new problems, but they certainly would slow the process and probably result in a less than optimum solution from the overall, long-term view.

To concurrently (or simultaneously) consider numerous interdependent design issues, in order to optimize within constraints, requires an application-specific model and considerable computing power. The thrust of CACED is to develop appropriate models that allow designers to

quickly establish and judge alternatives, simultaneously evaluating the compromises between often conflicting requirements.

Computer hardware is readily available to perform design evaluation tasks--the challenge lies in establishing appropriate equations and a framework in which they are to be effectively used. This report explains CACED structure, illustrates a practical application to refrigeration system design, and suggests areas of further study.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 PROBLEM STATEMENT

System design for satisfactory performance and minimum lifetime cost encompasses various engineering specialties. These specialties include "ilities" such as reliability, maintainability, operability, produceability, testability, and supportability, in addition to various aspects of electrical, mechanical or computer engineering. While the latter three disciplines provide the wherewithal to design many kinds of functional systems, the system's utility may be achieved at prohibitive production, operational, or support costs.

What approaches can be taken to achieve reasonable compromise between conflicting design requirements? The prevalent approach is that of "design audit." In this case, a design review meeting is held between the designer(s), "ilities" specialists (also called Specialty Engineers), and representatives of the more traditional engineering areas. This review provides feedback to the designer(s), but often is the first occasion for the various engineers to interact. In this way all pertinent design areas are addressed, but not as quickly or as well as they could be. Clearly, an early and active role for all participants would result in faster identification and probably better resolution of tradeoffs. Simultaneous consideration of all aspects of design, with feedback on "bottom line" factors, such as mean time between failures (MTBF) and cost, is known as Concurrent Engineering.

When various engineering specialties are involved, many interdependencies will exist. In order to evaluate design alternatives, these interdependencies must be identified and quantified. Generally, the resulting calculations cannot be done economically by hand. Therefore, computer programs are needed to manage the computing duties. Also, the computer provides a convenient medium for data entry/storage and display of results. The use of computers in such applications is known as Computer-Aided Concurrent Engineering Design (CACED).

2.0 COMPUTER-AIDED CONCURRENT ENGINEERING DESIGN (CACED)

Virginia Tech's understanding of CACED is primarily due to RAMCAD (Reliability And Maintainability in Computer-Aided Design) research sponsored by the USAF and performed by the Federal Systems Group of TRW and Virginia Tech. [1][2]

A top-level block diagram for the CACED process is given in Figure 1. The five blocks are described as follows:

Block 1--Human Designer(s)

The person(s) performing the design effort access the program and receive feedback through the human-computer interface (HCI), which normally consists of monitor and keyboard.

Block 2--Computer-Aided Design/Engineering Tools

CAD/E tools are software such as: FORTRAN, computer operating systems, modeling/analysis programs, etc..

Block 3--Estimators/Predictors

This part of the process accesses design inputs from block two and data from block five to determine values for design-dependent parameters (Y_d), such as reliability and design life. Such parameters may be subject to policy (as well as physical) constraints.

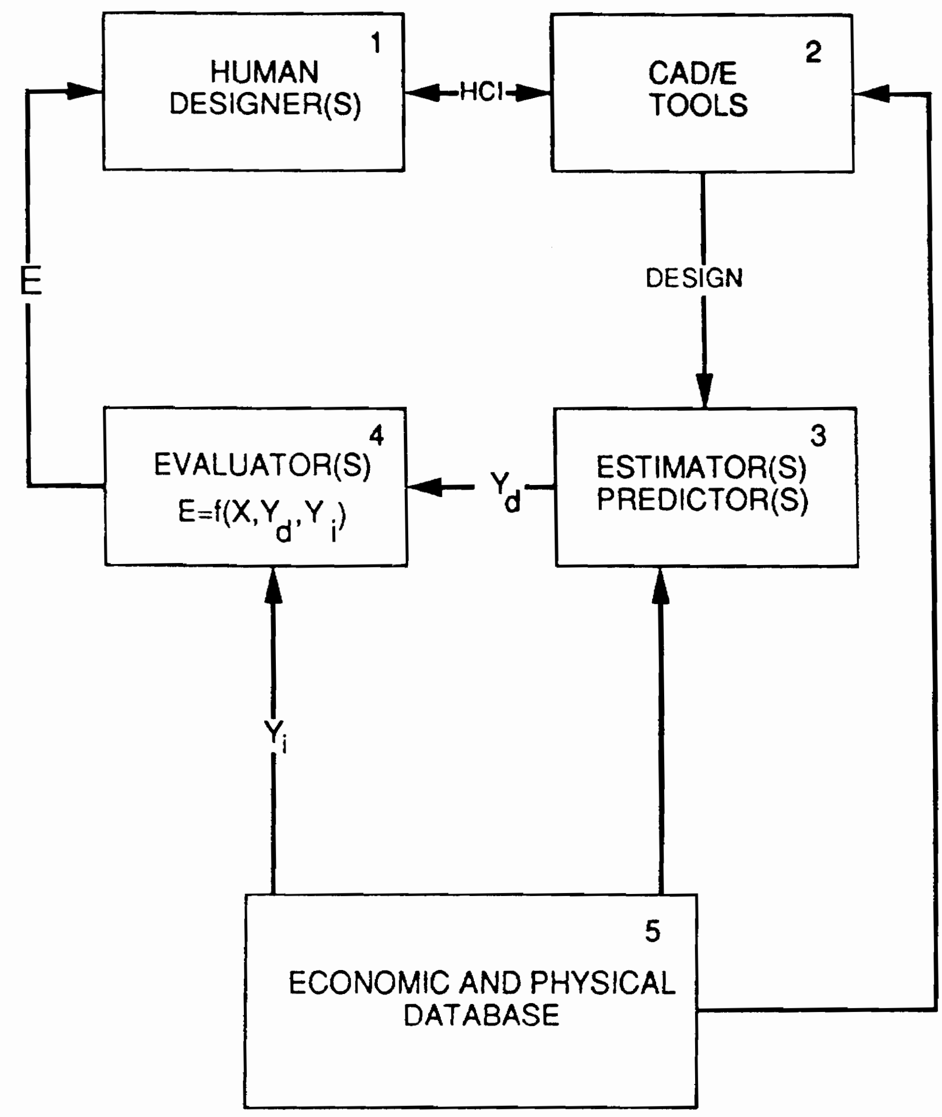


FIGURE 1--CACED MORPHOLOGY

Block 4--Evaluator

The evaluator program uses appropriate values of design-dependent and design-independent parameters (Y_i--see Block 5) in an application-specific effectiveness (E) function that returns optimum design variable values (X), such as minimum number of units to meet demand, and the E-value(s) for the current input data set. The measure of effectiveness would normally be cost (with effect of cash flow timing), but may be reliability, availability, accuracy, size, lifetime, etc.. Output of the E-value(s) from block two, if formatted properly, can be very useful in comparing competing design alternatives when more than one measure of effectiveness is to be considered.

Block 5--Economic and Physical Database

These design-independent parameters are for use in the estimation/prediction of Y_d values. They are also accessed for design evaluation. Types of data may include labor rates, material costs, interest rates, material densities, thermal conductivities, thermodynamic properties of various fluids, etc..

This report illustrates the CACED process for the large scale design of a refrigeration system, designated RE. By large scale, it is meant that a specified cooling demand (expressed in BTU/minute) is great enough and/or the spaces to be cooled are sufficiently separated that several individual units are required. Practical applications for such cases may be hospitals, industrial complexes, or equipment shelters.

It is assumed that all units will be identical, and that they will be repairable. Therefore, the design task requires an optimized determination of number of number of units to build, number of repair channels to provide, and the operational age at which to retire individual units. These determinations are made through use of a program (developed at Virginia Tech) called Repairable Equipment Population System (REPS). REPS employs finite queueing and optimizing techniques discussed in chapter twenty of [2].

3.0 REFRIGERATION DESIGN APPLICATION

The first steps in development of a CACED tool for refrigeration system design are requirements analysis and concept formulation. Qualitative top-level requirements are taken from the previous discussion. The conceptual design, from which detailed features evolve, is as follows:

Cooling is to be accomplished via heat exchangers (condenser and evaporator) using Freon-12 as a working fluid. The Freon is to be compressed prior to entering the condenser, and throttled between condenser and evaporator. Heat transfer is to be enhanced by forcing air over the condenser and evaporator fins. The compressor is to be a piston and cylinder type. Therefore, the refrigeration system consists of four major components (compressor, evaporator, condenser, and throttle) as shown in Figure 2. The Freon absorbs heat in the evaporator and rejects heat in the condenser. Separate fans are used to force air over condenser and evaporator fins.

Based on the above description, twenty-one input variables (design features), with associated outputs, have been identified, as shown in Figure 3. Note that system design is not strongly influenced by throttle characteristics. Therefore, no throttle-related inputs are made available to the designer.

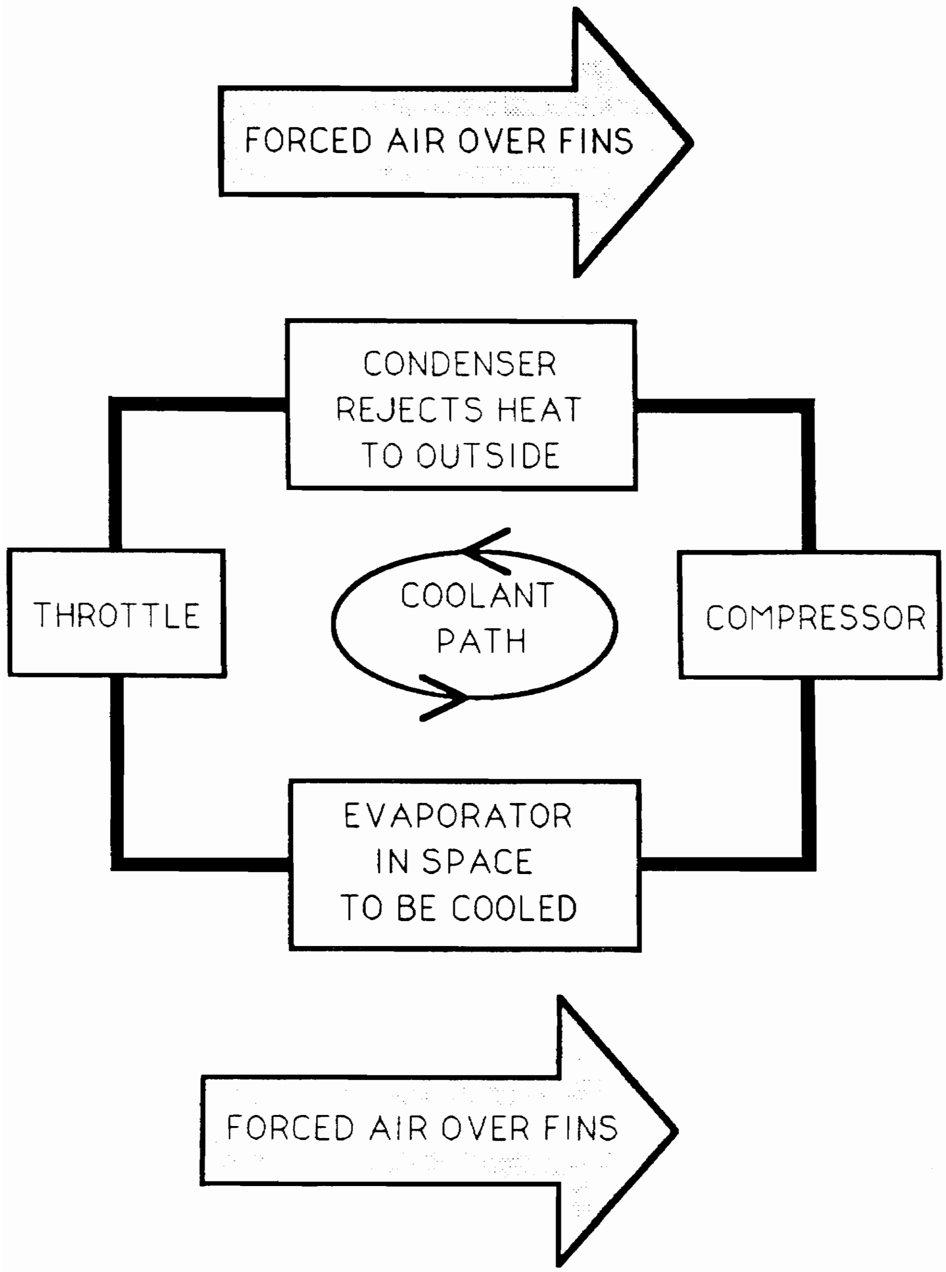


FIGURE 2 -- REFRIGERATION SYSTEM SCHEMATIC

	EVAPORATOR	COMPRESSOR	CONDENSER
INPUTS	INLET AIR TEMP EXIT AIR TEMP AIR VELOCITY FIN WIDTH FIN THICKNESS FIN PITCH TUBE RADIUS MEAN FREON TEMP	NUMBER OF CYLS CRANK SPEED PISTON DIAMETER MAX CYL VOLUME CLEARANCE VOL	INLET AIR TEMP EXIT AIR TEMP AIR VELOCITY FIN WIDTH FIN THICKNESS FIN PITCH TUBE RADIUS MEAN FREON TEMP
OUTPUTS	HEAT RATE NUMBER OF FINS TUBE LENGTH FAN HORSEPOWER	MOTOR HORSEPOWER	HEAT RATE NUMBER OF FINS TUBE LENGTH FAN HORSEPOWER

FIGURE 3 -- MAJOR COMPONENT INPUT/OUTPUT SUMMARY

Evaluating a system design for such a large number of inputs requires computation that cannot be done conveniently by hand. Therefore a computer program was written (in FORTRAN) for this application.[3]

This program is named RE (for REfrigeration) and is described here in the context of the CACED morphology previously outlined. (Most of the work in the development of this design application and computer program was done in a parallel effort to this report, and resulted in a paper [4] that explains CACED concepts within the system life-cycle progression of design, production, and support.)

Program features correspond to Figure 1 as follows:

Block 1--Human Designer(s)

The human-computer interface (HCI) is accomplished through keyboard and monitor. Through menu-driven selections, the user can access, modify, and store designs (input data sets).

Block 2--CAD/E Tools

CAD/E tools are computing power and other features of FORTRAN along with menu, input/output (I/O), sensitivity and summary screens that make information readily available and easily useable for the designer. In the program, these screens include input value restrictions, I/O status, a design variable and parameter summary, thermodynamic state diagrams, design feature sensitivity, and menu selections for file operations.

Block 3--Estimators/Predictors

These form a large computational part of the program, using the design features (inputs) from block two and the design-independent variables of block five in a series of calculations that provide design-dependent outputs to block four. These include, among others, motor horsepowers and heat exchanger tube lengths. The most pertinent variables are included in the I/O screen, Figure 4. For explanations of the relations developed for these variables, see page 14.

Block 4--Evaluators

The "bottom-line" evaluation measure is money. The estimate of total cost is most often used, and in this program it is converted into Equivalent Annual Expected Life-Cycle Cost (EAELCC). This takes into account interest rates and nonuniform flow of money in order to output a single value for comparison of alternatives. The REPS program is a major contributor to the design evaluating capability of RE. REPS outputs, along with certain values of Y_d and Y_i , are shown in Figure 5.

Refrigeration System Designer

Number of Cylinders	3.0000	Evap Heat Rate, B/m	6752.6
Crank Speed, RPM	3600.0	Cond Heat Rate, B/m	7356.3
Piston Diameter, In	2.5000	No of Evap Fins	.15706E+07
Maximum Cylinder Volume, Cu In	15.000	No of Cond Fins	.87946E+06
Cylinder Clearance Volume, Cu In	2.0000	Length Evap, In	.23559E+07
Condenser Mean Temperature, Deg F	100.00	Length Cond, In	.43973E+06
Condenser Inlet Air Temp, Deg F	85.000	Compressor Motor Hp	15.000
Condenser Exit Air Temp, Deg F	90.000	Evap Fan Motor Hp	.80000E-02
Evaporator Inlet Air Temp, Deg F	65.000	Cond Fan Motor Hp	.80000E-02
Evaporator Exit Air Temp, Deg F	60.000	Acquisition Cost $	1627.5
Condenser Fin Width, In (C)	2.0000	Design Life, Hr	.12246E+06
Condenser Fin Thickness, In (C)	.60000E-01	MTTR, Hr	309.56
Condenser Fin Pitch, In (C)	.50000	Overall MTBF, Hr	155.62
Condenser Tube Radius, In (C)	.12500	Expected Shortage	4.6908
Condenser Air Velocity, fpm (C)	5.0000		
Evaporator Fin Width, In (E)	2.0000	Expected Cooling, B/m	8840.3 0
Evaporator Fin Thickness, In (E)	.60000E-01		
Evaporator Fin Pitch, In (E)	1.5000	Expected Annual Equiv.	0
Evaporator Tube Radius, In (E)	.12500	Life Cycle Cost $	.12541E+07
Evaporator Air Velocity, fpm (E)	10.000		
Evaporator Mean Temperature, Deg F	60.000	Enter Data	

F1 = T-S F2 = P-H F3 = P-V

FIGURE 4 -- INPUT/OUTPUT SCREEN

Parameter/Variable Review		
Cooling Demand, Btu/min	40000	
Interest Rate, %	10.000	DESIGN INDEPENDENT PARAMETERS
Shortage Penalty, $/(Btu/min)	30.800	
Annual Repair Facility Cost, $	5000.0	
Repair Facility Annual Operating Cost, $	10000.	
Unit Acquisition Cost, $	2224.4	DESIGN DEPENDENT PARAMETERS
Salvage Value, $	.00000	
Design Life, Yrs	13.000	
Annual Fuel Cost, $	9986.2	
Labor Cost, $	.15849E+06	
Other Operating Costs, $	.00000	
Number of Units Needed to Meet Demand	6.0000	
Number of Units Deployed	6.0000	DESIGN VARIABLES
Number of Repair Channels	1.0000	
Economic Life, Yrs	13.000	

FIGURE 5 -- PARAMETER/VARIABLE REVIEW SCREEN

Block 5--Physical and Economic Database

The greatest part of this database is composed of tabular files of thermodynamic values for the refrigerant and for air. There is a table of motor horsepowers (for fans and compressor) whose values are used in cost and performance determinations. Also, there are values for unit costs for various types of labor, material density, etc. that are built into equations for material, manufacturing, and maintenance costs. No economic database exists as such (e.g. time value of money tables found in engineering economy and finance textbooks). These factors are computed rather than stored.

3.1 Estimator/Predictor Relations

Following are brief descriptions of several design variables used in RE. Source code and line for each are given in Appendix A.

Compressor Cost

It is assumed that the compressor is the piston and cylinder type and that cylinders are arranged in a single line. Piston/cylinder count and geometric inputs are used to establish compressor size, weight, and surface area (pistons and cylinders) to be machined. Material, machining and assembly costs are combined to determine compressor cost.

Note that the motor required to drive the compressor is not included here.

Motor Horsepowers and Costs

Separate motors are required for each fan. Another motor is required to drive the compressor, as described above. Power requirements for these motors come from thermodynamic and heat transfer relations. Inputs used include those of heat exchanger geometry, coolant temperatures, and air velocities. For each motor, the program selects the smallest standard size (horsepower table is built in) that will meet the requirement. Each power value is used in a straight-line equation to estimate cost. This costing equation is the result of a least squares fit of recent manufacturers' data.

Design Life

Unit design life is assumed to be dependent on the compressor only, due to numerous moving parts and high outlet temperature. Design life is assumed to be inversely proportional to RPM, and includes a multiplier that halves the result for a twenty degree Fahrenheit rise in outlet temperature. The basis for this temperature dependency is taken from [5].

Mean Time To Repair (MTTR)

This is assumed to be proportional to the number of compressor cylinders and to the total length of heat exchanger tubes. (Tube lengths are from fin pitch inputs multiplied by number of fins for required heat transfer.) In this way, repair time, and therefore maintenance costs, is higher for more complex compressor designs and for larger heat exchangers. Compressor repair would consist of teardown, parts replacement, and reassembly. Heat exchanger repair is limited to cleaning fouled fins and tubes.

Unit Mean Time Between Failures (MTBF)

This is predicted to be inversely proportional to the number of compressor cylinders and compressor RPM. Motors are considered to be failure-free due to the assumption that they run well below capacity (see explanation under MOTOR COSTS). It is further assumed that heat exchangers do not fail, but that they do contribute to MTTR as described above. Justification is that heat exchangers are to be cleaned whenever a unit is down for compressor repair.

Preventive Maintenance (PM)

The cost of PM is assumed to be proportional (and due entirely) to total heat exchanger tube length. PM is separate from repair efforts previously discussed. It includes efforts (such as exterior cleaning, addition of anti-fouling chemicals, or recharging of Freon) that do not result in system downtime.

Acquisition Cost Per Unit

This is the cost of all motors, plus compressor and heat exchangers. Motor and compressor costs have been previously discussed. Heat exchanger cost is based only on the weight (multiplied by unit cost) of Aluminum for fins and tubes. This is because no conditions have been set for size or style. Note also that ducting is not considered.

Fuel Cost

All power is electric, at a cost of $0.08/kilowatt-hour.

Labor Cost

These costs are for repairs only. PM costs are as above.

Salvage Value

Set at zero at design life. For economic life shorter than design life, straight-line depreciation is applied.

4.0 REFRIGERATION DESIGN EXAMPLE

Consider an example refrigeration design problem where the requirements have been established as:

Minimum Capacity	--	40 KBtu/min
Maximum EAELCC	--	$250,000
Minimum Overall MTBF	--	100 hours

Baseline values for the twenty-one design features are shown in the left side of Figure 4. Results are shown on the right side of this I/O screen. It is seen that the baseline design meets the MTBF requirement, but not those for capacity or cost. The designer's task is to improve on this baseline by altering the values of any or all design features.

Five aids to the designer are available, three of which should be particularly useful. These three are:

1) Input/Output Screen

This gives all inputs and several outputs, and is the vehicle for altering the design features. An added capability is display of comparative result indicators. In the right side of this screen (Figure 4), results for capacity and cost are accompanied by positive, negative or zero signs. These signs indicate the direction of change for these results as compared to those for the previous set of design features.

(Note that for the initial set, the indicators default to zero.)

2) **Parameter/Variable Summary Screen**

This screen gives values of design-independent parameters along with several resulting design-dependent parameters and design variables. Baseline values for this screen are shown in Figure 5.

3) **Sensitivity Analysis Screen**

This screen is considered to be the most useful for making decisions regarding changes to the design features. It is developed and utilized as described in the following section.

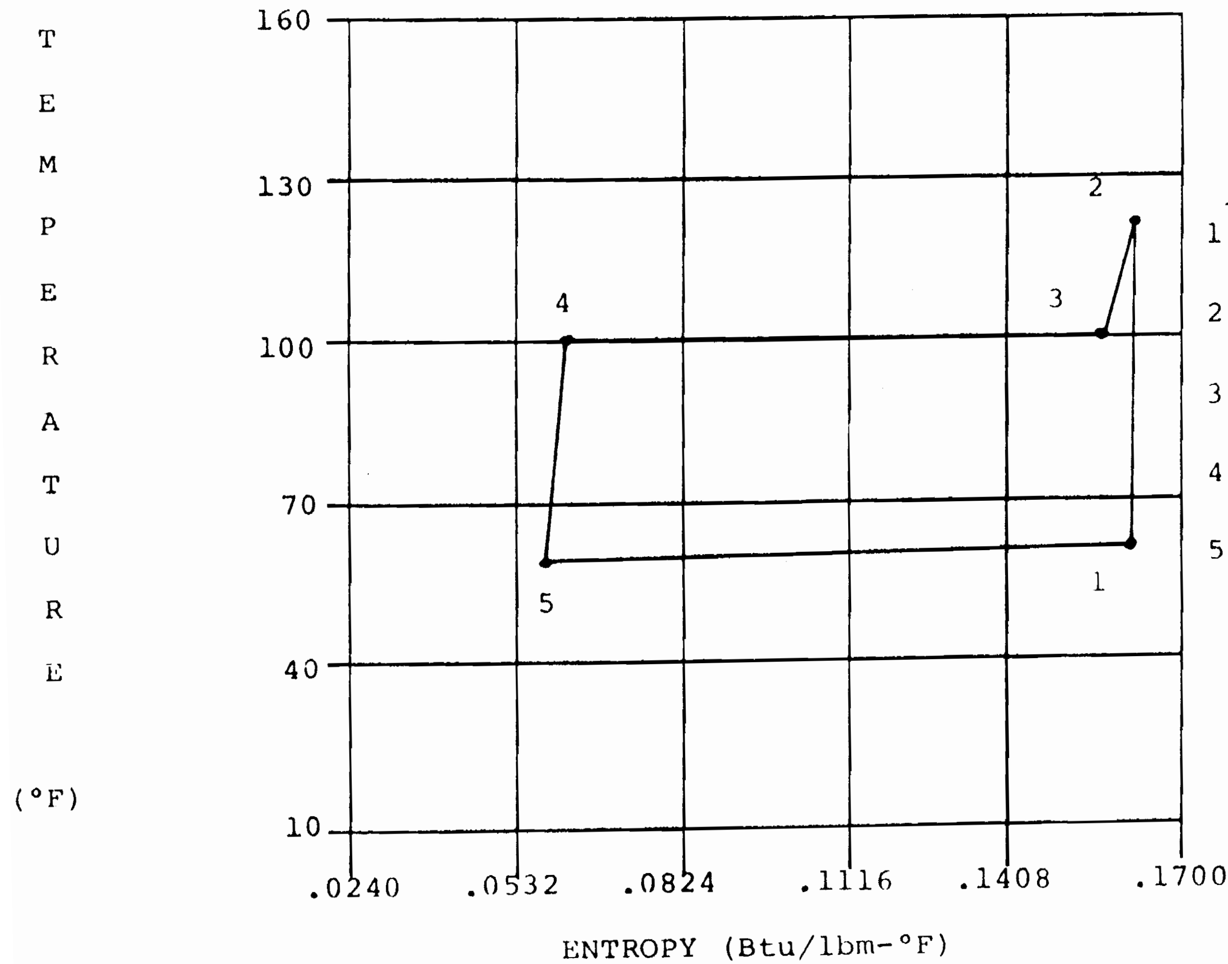
In addition to the above, the designer is provided with a list of design feature constraints imposed by the program (see Figure 6) and may access graphical displays of thermodynamic state values (e.g. the baseline case for Temperature--Entropy as shown in Figure 7).

DESIGN FEATURE INPUT VALUE CONSTRAINTS

DESIGN FEATURE	CONSTRAINT
Number of Cylinders	>= 1
Crank Speed, RPM	900, 1200, 1800, or 3600 RPM
Piston Diameter, In	>= 1
Maximum Cylinder Volume, Cu In	>= 1.5
Cylinder Clearance Volume, Cu In	>= 1
Condenser-Mean Temperature, Deg F	> EVAPORATOR TEMPERATURE
Condenser, Inlet Air Temp, Deg F	< CONDENSER EXH. AIR TEMP. & >=-457
Condenser, Exit Air Temp, Deg F	> CONDENSER INLET AIR TEMPERATURE
Evaporator, Inlet Air Temp, Deg F	> EVAPORATOR EXH. AIR TEMPERATURE
Evaporator, Exit Air Temp, Deg F	>= -457
Condenser Fin Width, In (C)	> 2*CONDENSER TUBE RADIUS
Condenser Fin Thickness, In (C)	>= .02
Condenser Fin Pitch, In (C)	>= CONDENSER FIN THICKNESS +.1
Condenser Tube Radius, In (C)	>= .125
Condenser Air Velocity, fpm (C)	> 0
Evaporator Fin Width, In (E)	> 2*EVAPORATOR TUBE RADIUS
Evaporator Fin Thickness, In (E)	>= .02
Evaporator Fin Pitch, In (E)	>= EVAPORATOR FIN PITCH + .1
Evaporator Tube Radius, In (E)	>= .125
Evaporator Air Velocity, fpm (E)	> 0
Evaporator-Mean Temperature,Deg F	> -457

HIT F1, F2 OR ESCAPE TO CONTINUE

FIGURE 6 -- INPUT CONSTRAINTS SCREEN



	TEMP.	ENTROPY
1	60.00	0.1647
2	119.06	0.1647
3	100.00	0.1631
4	100.00	0.0682
5	60.00	0.0641

FIGURE 7 -- TEMPERATURE-ENTROPY DIAGRAM

4.1 Sensitivity Analysis

The sensitivity subroutine is an option that greatly speeds the design process.[6] It identifies the relative effect on EAELCC of each input variable for the current configuration, leading the designer directly to the highest impact inputs. "Sensitivity" is determined from the EAELCC change that would be introduced by a fixed percentage change (typically a 5% increase) in a given input. The actual calculation takes the EAELCC change and divides by 1.05, which is the ratio of "new" input value to "old" value. Sensitivity values are much like partial derivatives in that, for each calculation, twenty of twenty-one input variables are held constant.

For each of the twenty-one design features, the subroutine changes the value, recalculates EAELCC, determines and saves sensitivity, and returns the input to its' previous value. Once all sensitivities are determined, they are ranked by absolute values. Then, output to the screen are the names and values of all inputs, with rankings (one through twenty-one) and direction (positive, zero or negative). An example of this screen is given in Figure 8.

The divisor 1.05 is used to treat unlike inputs in a consistent manner (i.e. a one inch change in fin size cannot be compared to a one RPM change in compressor speed).

Design Feature Sensitivity			
Number of Cylinders	3	3.0000	+
Crank Speed, RPM	7	3600.0	+
Piston Diameter, In	19	2.5000	+
Maximum Cylinder Volume, Cu In	6	15.000	+
Cylinder Clearance Volume, Cu In	1	2.0000	+
Condenser-Mean Temperature, Deg F	2	100.00	+
Condenser, Inlet Air Temp, Deg F	8	85.000	+
Condenser, Exit Air Temp, Deg F	21	90.000	-
Evaporator, Inlet Air Temp, Deg F	5	65.000	-
Evaporator, Exit Air Temp, Deg F	20	60.000	+
Condenser Fin Width, In (C)	12	2.0000	-
Condenser Fin Thickness, In (C)	18	.60000E-01	-
Condenser Fin Pitch, In (C)	13	.50000	+
Condenser Tube Radius, In (C)	17	.12500	-
Condenser Air Velocity, fpm (C)	15	5.0000	-
Evaporator Fin Width, In (E)	9	2.0000	-
Evaporator Fin Thickness, In (E)	16	.60000E-01	-
Evaporator Fin Pitch, In (E)	10	1.5000	+
Evaporator Tube Radius, In (E)	14	.12500	-
Evaporator Air Velocity, fpm (E)	11	10.000	-
Evaporator-Mean Temperature,Deg F	4	60.000	+

FIGURE 8 -- DESIGN FEATURE SENSITIVITY SCREEN

Because 1.05 is actually a RATIO of "new to old" input values, the denominator in the sensitivity calculation is dimensionless and the results are comparable. The only exceptions to the 1.05 divisor occur when the number of cylinders (NCY) or compressor RPM are being changed. Because fractional values of NCY are illogical, the divisor is taken as (NCY+1)/NCY. For compressor RPM, because only four input values are allowed, the divisor is an appropriate ratio of permissible values.

Although this option is a great aid in quickly arriving at an optimal set of inputs, a word of caution is in order. The designer must be alert to trends while searching through the design space. Due to the 5% input changes used by the subroutine, it will "step-over" relative maximum or minimum points when the input being changed is sufficiently close to such a point. Whether near a maximum or a minimum, the sensitivity determined is subject to large error. Particularly bad would be a case where the designer finds artificially low sensitivity for a particular input and never elects to vary that value. The most logical approach in using this subroutine is to experiment with the lowest as well as the highest-ranked inputs.

4.2 Design Example Results Discussion

Utilizing the design aids previously discussed maintaining an awareness of the limitations of the sensitivity routine), the designer may arrive at a data set similar to that of Figure 9. It is seen that the performance requirement is now met, with dramatic cost savings. However, there has been a compromise to MTBF. Considering the cost savings and the projected cooling capacity for the alternative, this design may be considered worthy. This assumes that there is no compelling reason not to waive the MTBF requirement, and is reasonable because expected cooling is closely related to MTBF. The designer may, at this point, elect to choose the better alternative or continue to search for improvement. If results are sufficiently stable as inputs with high sensitivity rankings are manipulated, the designer may be content that a near-optimal set of design features has been attained.

The designer may also find useful a graphic comparison between competing alternatives. A display similar to Figure 10 would be appropriate. Vertical axes indicate changes in performance, availability, etc. between the baseline and alternative cases. The horizontal axis shows EAELCC for each case. With such a format, design compromises are quantified and easily understood.

Refrigeration System Designer

Number of Cylinders	1.0000	Evap Heat Rate, B/m	3226.7
Crank Speed, RPM	3600.0	Cond Heat Rate, B/m	3515.2
Piston Diameter, In	2.5000	No of Evap Fins	.13593E+06
Maximum Cylinder Volume, Cu In	20.000	No of Cond Fins	49423.
Cylinder Clearance Volume, Cu In	2.0000	Length Evap, In	25827.
Condenser Mean Temperature, Deg F	100.00	Length Cond, In	9390.4
Condenser Inlet Air Temp, Deg F	85.000	Compressor Motor Hp	7.5000
Condenser Exit Air Temp, Deg F	90.000	Evap Fan Motor Hp	.25000E-01
Evaporator Inlet Air Temp, Deg F	65.000	Cond Fan Motor Hp	.25000E-01
Evaporator Exit Air Temp, Deg F	60.000	Acquisition Cost $	834.63
Condenser Fin Width, In (C)	4.0000	Design Life, Hr	.12246E+06
Condenser Fin Thickness, In (C)	.90000E-01	MTTR, Hr	13.522
Condenser Fin Pitch, In (C)	.19000	Overall MTBF, Hr	62.267
Condenser Tube Radius, In (C)	.19000	Expected Shortage	.21455E-01
Condenser Air Velocity, fpm (C)	50.000		
Evaporator Fin Width, In (E)	4.0000	Expected Cooling, B/m	45104. -
Evaporator Fin Thickness, In (E)	.90000E-01		
Evaporator Fin Pitch, In (E)	.19000	Expected Annual Equiv.	-
Evaporator Tube Radius, In (E)	.19000	Life Cycle Cost $	.14981E+06
Evaporator Air Velocity, fpm (E)	50.000		
Evaporator Mean Temperature, Deg F	60.000	Enter Data	

F1 = T-S   F2 = P-H   F3 = P-V

FIGURE 9 -- ALTERNATIVE DESIGN DATA SET

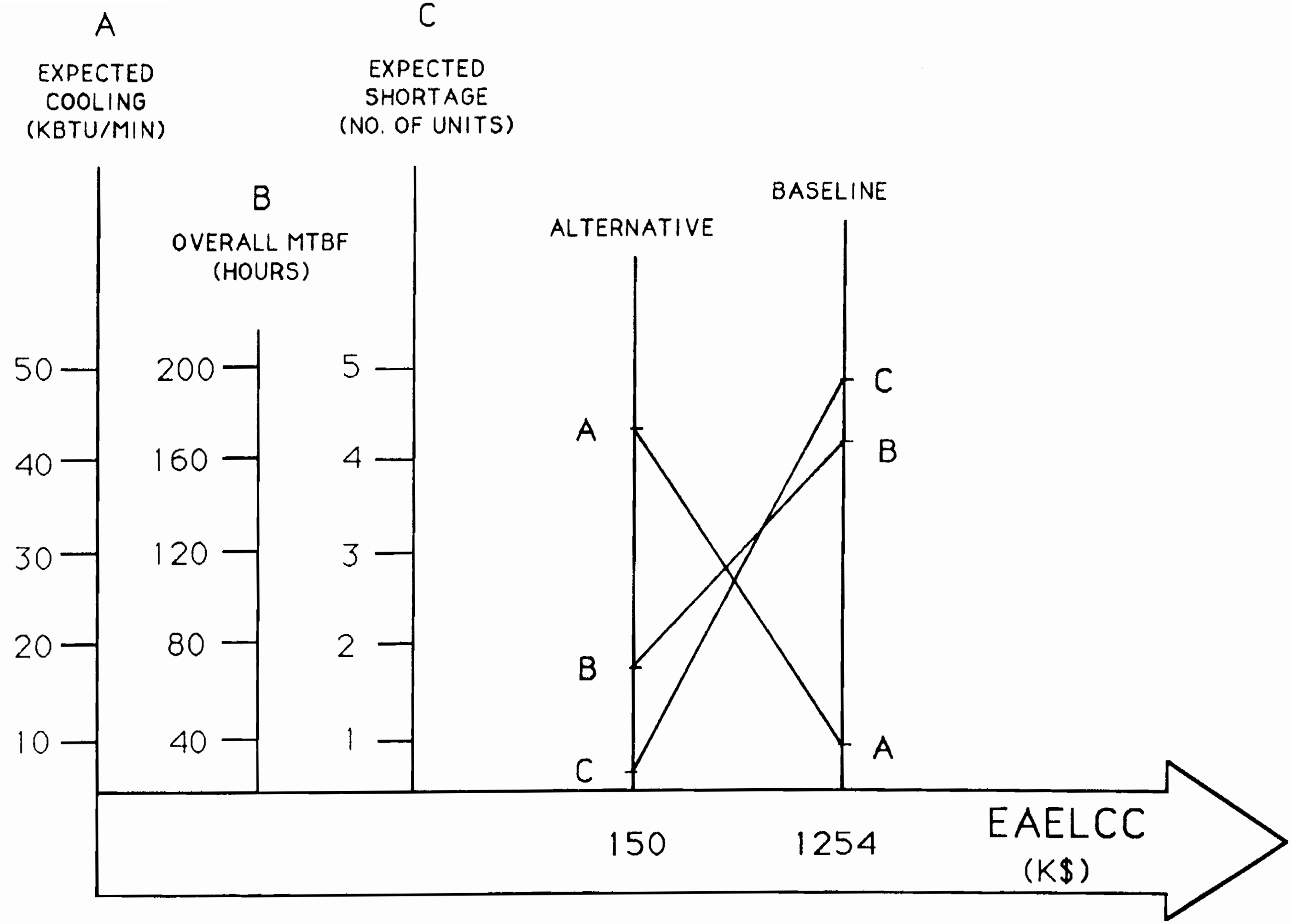


FIGURE 10 -- DESIGN COMPARISON

5.0 SUGGESTIONS FOR FURTHER STUDY

Although the program structure is very useful, certain estimators could be more realistic. For example, compressor MTBF and MTTR result from reasonable assumptions, but these do not have firm backup data. MTBF and MTTR for mechanical assemblies are much more difficult to determine than for electronic assemblies. This is due to the multitude of potential failure types and repair methods for a given mechanical device. This is a problem recently studied by Eagle Technologies.[7]

There are other relations that do not currently exist in the program, but which may be useful. Suggested areas for further development are:

1) Heat exchanger performance degradation with time, due to fouling of fins and tubes. Effect of an (assumed to be) optimal maintenance schedule may be evaluated against that of an automatic performance monitor.

2) Dependencies of convective heat transfer coefficients on air speed, fin geometry (e.g., tapered fins are known to perform better than rectangular, but are more expensive to produce), and fin pitch (e.g., boundary layer buildup). Currently, dependencies on shape and pitch are omitted, except that pitch inputs are

constrained to provide at least one-tenth of an inch of space between fins.

3) Comparison of refrigerants. Only R-12 is currently considered. Improvement might be found with another coolant, such as ammonia or R-22, but much more computer memory and run-time would be required.

4) Inclusion of air ducts. Current assumption is that none exist. Each air conditioner behaves as a "window unit."

5) Installation costs (which are not the same as acquisition costs) have not been considered.

6) Failure rate (MTBF) is currently treated as a single value that is independent of equipment age. It is more realistic to assume higher failure rates early and late in equipment life, with a constant value for the middle years. This is due to "burn-in" failures early and "wear-out" failures late in the equipment lifetime. This is explained in greater detail in [8].

7) Salvage value at the end of equipment design life is currently assumed to be zero. However, it may be reasonable to develop a relation that establishes this value as a function of equipment complexity (acquisition cost).

8) Due to the probabilistic nature of the REPS routine, some finite value of expected number of units short must exist, regardless of the design feature values. However, this does not necessarily mean that expected cooling will be less than the stated demand. (Because shortage is taken against the number of units deployed, rather than the minimum required to meet demand.) As currently written, the program applies a monetary shortage penalty, even when expected cooling exceeds demand. Although this penalty should be relatively low for a good design, it would be logical to exclude in such a case.

9) RE could utilize a program that outputs a visual comparison of competing alternatives, similar to the format of Figure 10. The framework for such a program now exists [9] which could be linked to RE as detailed in Appendix B.

REFERENCES

[1] Fabrycky, W.J., Virginia Tech Project Director, RAMCAD Software Development, Contract No. F33615-87-C-0002, USAF/HRL, Wright-Patterson AFB, Ohio.

[2] SYSTEMS ENGINEERING AND ANALYSIS, 2nd edition, Blanchard, B.S. and Fabrycky, W.J., Prentice-Hall, Inc., 1990.

[3] RE, a PC-based program, was developed by Graduate Research Assistant M. Feng under the direction of Professor R.G. Mitchiner. Graduate Research Assistant J. Jennings made a contribution to the evaluation code. Project support came from the Virginia CIT and IBM under grant no. CIT/CAE 86-008, W.J. Fabrycky and R.G. Mitchiner co-principal investigators.

[4] CONCURRENT MECHANICAL SYSTEM DESIGN: A COMPUTER-AIDED DEMONSTRATION was co-authored by J. Altenhof, W. Fabrycky, M. Feng, and R. Mitchiner in 1989. This paper was presented by R. Mitchiner on October 25, 1989 at the First Annual Symposium on Mechanical System Design in a Concurrent Engineering Environment (held at the University of Iowa, Iowa City, IA) and is included in the proceedings.

[5] Ambrose, E.R., Heat Pumps and Electric Heating, John Wiley and Sons, Inc., 1966.

[6] The Sensitivity subroutine was rewritten by J. Altenhof to calculate "derivative" values and feed these values into a previously existing sorting routine for ranking. This screen format was adapted to include sign of "derivative."

[7] (DRAFT) HANDBOOK OF RELIABILITY PREDICTION PROCEDURES FOR MECHANICAL EQUIPMENT, Eagle Technology, Inc., Arlington, VA, in conjunction with the Product Assurance and Engineering Directorate of the Belvoir Research, Development, and Engineering Center, Fort Belvoir, VA, 1988.

[8] The reliability "bathtub" curve is discussed in chapter thirteen of [2].

[9] DESIGN EVALUATION DISPLAY, a program developed at Virginia Tech in 1989 by Graduate Research Assistant J. Jennings under the direction of Professor W.J. Fabrycky, outputs graphic screens similar to Figure 10 and can be adapted to take inputs directly from RE.

APPENDIX A

ESTIMATOR/PREDICTOR SUMMARY

The estimators/predictors discussed in 3.1 are found in RE source code, with names and line numbers as listed below. Each is determined within subroutine COMPUTE. The values are passed to subroutine XOPT, which invokes REPS (discussed in 3.0) and returns optimum values of design variables and EAELCC (see Figures 5 and 4, respectively).

ESTIMATOR/ PREDICTOR	VARIABLE NAME	LINE NUMBER
Compressor Cost	RESULTS(10)	1792
Compressor Motor HP	RESULTS(7)	1809
Evaporator Fan HP	RESULTS(8)	1820
Condenser Fan HP	RESULTS(9)	1827
Motor Costs (built into Unit Acquisition Cost--see below)	---	1882
Design Life	RESULTS(11)	1848
MTTR	RESULTS(12)	1851
Unit MTBF	RESULTS(13)	1852
PM Cost	AOPPM	1892
Unit Acquisition Cost	ACQ	1882
Fuel Cost	AOPFU	1890
Labor Cost	AOPLA	1891
Salvage Value	SALV	1883

APPENDIX B

RE INPUTS TO DESIGN EVALUATION DISPLAY PROGRAM

The Design Evaluation Display program referenced in 5.0 can be adapted to take inputs directly from RE. Inputs to this program would be as listed below:

INPUT	NAME	LINE NUMBER
Expected Cooling	EXCOOL	1915
Overall MTBF	RESULTS(14)	1916
Expected Shortage	ES	3066
EAELCC	CALCUL	3083

APPENDIX C

RE SOURCE CODE

The source code that follows is written in Fortran. Of particular interest are the following subroutines, listed in order of appearance, and accompanied by brief reviews.

MENU

Provides the means to access design editing, file, and CAD/E operations.

EDIT

Used to change input data values, and includes error trapping. Prevents entry of physically impossible values, such as temperatures below absolute zero, or illogical values such as fin thickness less than .02 inch. EDIT also contains code that generates the input/output screen.

COMPUTE

Performs thermodynamic and heat transfer calculations. Contains design-independent parameter values, such as motor horsepower, refrigerant enthalpy, etc. Determines design dependent parameter values (e.g. evaporator tube length) via estimating/predicting relationships, several of which are described in 3.1. Outputs values of high interest to EDIT's input/output screen.

REVIEW

Displays design-independent parameter values, such as interest rate, shortage penalty, etc., design-dependent parameter values, such as annual fuel and labor costs, and the design variable values for: minimum number of units to meet demand, optimum numbers for deployment, repair channels, and equipment retirement.

SENSITIVE

Uses EAELCC values as explained in 4.1 to point the designer to the design features with the greatest impact on cost.

XOPT

Determines optimum design variable values (taking inputs from COMPUTE) by invoking REPS. Ultimately returns Equivalent Annual Life-Cycle Cost through FUNCTION CALCUL.

CALCUL

Determines expected number of units short (at any instant in time), and operates on this value, the REPS outputs, and certain results of COMPUTE to determine cost.

CONSTRAINTS

Displays input value limitations which were established in EDIT. This screen appears automatically once the designer has exited the initial banner screen, and remains until the designer requests MENU. The screen then cannot be accessed without exiting and restarting the program.

```
$DEBUG
$LARGE
C
C     MAIN PROGRAM
C
C   INITIALIZATION
C
C       PUT UP STARTING LOGO FROM BSAVED FILE
C
      REAL*4 DATA
      COMMON /CINP/ DATA(21)
C
C     1     NCY     NO OF CYLS
C     2     RPM     CRANK SPEED, RPM
C     3     DP      PISTON DIA, IN
C     4     VMAX    MAX VOL/CYL, CUBIC IN
C     5     VCL     CL VOL/CYL, CUBIC IN
C     6     T3      COND TEMP, DEG F
C     7     TCIN    COND INLET AIR TEMP, DEG F
C     8     TCOUT   COND EXH AIR TEMP, DEG F
C     9     TEIN    EVAP INLET AIR TEMP, DEG F
C    10     TEOUT   EVAP EXH AIR TEMP, DEG F
C    11     D       COND FIN WIDTH, IN
C    12     TH      COND FIN THICKNESS, IN
C    13     LPI     COND FIN PITCH, IN
C    14     R       COND TUBE RADIUS, IN
C    15     UH      COND AIR VELOCITY, IN/S
C    16     ED      EVAP FIN WIDTH,IN
C    17     ETH     EVAP FIN THICKNESS, IN
C    18     EL      EVAP FIN PITCH, IN
C    19     ER      EVAP TUBE RADIUS, IN
C    20     EUH     EVAP AIR VELOCITY, IN/S
C    21     T1      EVAP TEMP, DEG F
C
      DATA(1)=2.
      DATA(2)=1800.
      DATA(3)=3.
      DATA(4)=20.
      DATA(5)=2.
      DATA(6)=100.
      DATA(7)=80.
      DATA(8)=90.
      DATA(9)=70.
      DATA(10)=60.
      DATA(11)=1.
      DATA(12)=.06
      DATA(13)=1.
      DATA(14)=.125
      DATA(15)=10.
      DATA(16)=1.
```

```
      DATA(17)=.06
      DATA(18)=1.
      DATA(19)=.125
      DATA(20)=10.
      DATA(21)=60.
C
      REFCST=0.
      RFCOOL=0.
      BBTEST=0.
C     ***REFCST AND RFCOOL INITIALIZED BEFORE CHANGED IN SUBROUTINE
C         COMPUTE (VIA SUBROUTINE EDIT)***
C
C                 *       1 - COLOR TCSTXT   2 - MONO TCSTXT
C                   *     1 - CGA            2 - EGA
C     CALL TCSCFG(1,2)
C
C     CALL TCSIND(IARG)   -- FOR EGA
C                                IARG   -   SETS COLOR FOR TCSLIN,
C                                           TCSFON, ETC.
C
      CALL INITT(11)
C     CALL LOGO
C     LOGO CALLS VERSION ONE OF BANNER SCREEN
      CALL BANNER
C     BANNER CALLS VERSION TWO OF BANNER SCREEN
C     CALL PROTECT
      CALL BINITT
      CALL MENU
      END
C
      SUBROUTINE WAITTO
      CALL TCSCUR(0,6,7)
      CALL TCSTXT(21,71,0,192,8,'IMMMMMM;')
      CALL TCSTXT(22,71,0,192,8,': WAIT :')
      CALL TCSTXT(23,71,0,192,8,'HMMMMMM<')
      CALL WAITTNO
      RETURN
      END
C
      SUBROUTINE GETFKEY(IFKEY,KEYMIN,KEYMAX,ESC)
      LOGICAL ESC
      ESC=.FALSE.
C
C    GET FUNCTION KEY
C
10    CALL TCSKEY (ISCAN,IASC,NCHAR)
      IFKEY=ISCAN-58
C
C     BE SURE THAT IT IS A FUNCTION KEY
C     THEN BRANCH TO APPROPRIATE SET OF KEY VARIABLE
```

```
C
      IF(IASC.EQ.27) GO TO 100
      IF(IASC.EQ.27) IFKEY=0
      IF(NCHAR.NE.2) GO TO 9
      IF(IFKEY.GE.KEYMIN.AND.IFKEY.LE.KEYMAX) RETURN
C
C      WRONG KEY,  RING BELL AND WAIT
C
9     CALL TCSBEL
      CALL TCSBEL
      CALL TCSBEL
      GO TO 10
C
100   IFKEY=0
      ESC=.TRUE.
      RETURN
      END
C
      SUBROUTINE NOMODEL(IER)
C
C    PUT UP MESSAGE THAT NO MODEL EXISTS
C
      REAL*4 DATA
      COMMON /CINP/ DATA(21)
      IER=0
      DO 10 I=1,21
      IF(DATA(I).NE.0) RETURN
10    CONTINUE
      CALL TCSSCR(13,0,23,79,0,7)
      CALL TCSTXT(17,24,0,4,31,'VDDDDDDDDDDDDDDDDDDDDDDDDDDDDD7')
      CALL TCSTXT(18,24,0,4,31,':          NO MODEL EXISTS      :')
      CALL TCSTXT(19,24,0,4,31,'GDDDDDDDDDDDDDDDDDDDDDDDDDDDDD6')
      CALL TCSTXT(20,24,0,4,31,':    RETURNING TO MAIN MENU     :')
      CALL TCSTXT(21,24,0,4,31,'SDDDDDDDDDDDDDDDDDDDDDDDDDDDDD=')
      CALL TCSBEL
      CALL TCSBEL
      CALL TCSBEL
      CALL WAITTO
      IER=1
      RETURN
      END
C
      SUBROUTINE TCSIN3(IROW,ICOL,IATTRB,IATTRF,NO)
      CHARACTER*3 N1
      WRITE(N1,1) NO
1     FORMAT(I3)
      CALL TCSTXT(IROW,ICOL,IATTRB,IATTRF,3,N1)
      RETURN
      END
C
```

```
      SUBROUTINE TCSINT(IROW,ICOL,IATTRB,IATTRF,NO)
      CHARACTER*2 N1
      WRITE(N1,1) NO
1     FORMAT(I2)
      CALL TCSTXT(IROW,ICOL,IATTRB,IATTRF,2,N1)
      RETURN
      END
C
      SUBROUTINE TCSREAL(IROW,ICOL,IATTRB,IATTRF,X)
      CHARACTER*10 N1
      WRITE(N1,1) X
1     FORMAT(G10.5)
      CALL TCSTXT(IROW,ICOL,IATTRB,IATTRF,10,N1)
      RETURN
      END
C
      SUBROUTINE WAITTNO
      CALL TCSDLY(2.)
      RETURN
      END
C
      SUBROUTINE TCS2REAL(IROW,ICOL,IATTRB,IATTRF,X)
      CHARACTER*11 N1
      WRITE(N1,1) X
1     FORMAT(G11.5)
      CALL TCSTXT(IROW,ICOL,IATTRB,IATTRF,11,N1)
      RETURN
      END
C
      SUBROUTINE PROTECT
      RETURN
      END
C
      SUBROUTINE MENU
      CHARACTER*16 BUFF16
      CHARACTER*80 LBUFF
      CHARACTER*1 BUFF(80)
      REAL*4 DATA
      COMMON /CINP/ DATA(21)
      COMMON AATEST
      LOGICAL ESC
      CALL CONSTRAINTS
C     ****SUBROUTINE CONSTRAINTS GIVES INPUT ERROR TRAP VALUES***
1     CONTINUE
      CALL TCSMOD(0)
      CALL TCSBLK(0,0)
      CALL TCSCUR(0,6,7)
      CALL TCSTXT (0, 0,0,12,36,'REFRIGERATION SYSTEM DESIGNER V1.00 ')
      CALL TCSTXT (2, 0,0,11,9,'MAIN MENU')
      CALL TCSTXT (4, 0,0,9,36,'F1  Modify System Model              ')
```

```
      CALL TCSTXT (5, 0,0,9,36,'F2  File Operations - Store/Retreive')
      CALL TCSTXT (6, 0,0,9,38,'F3  Review Design Indep/Dep Parameters')
      CALL TCSTXT (7, 0,0,9,32,'F4  Perform Sensitivity Analysis')
      CALL TCSTXT (8, 0,0,9,8,'ESC Exit')
      CALL TCSTXT (10,0,0,11,25,'Tap A Key For Your Choice')
      CALL TCSTXT (23,0,0,0,0,' ')
      CALL TCSBLK(1,0)
C
C       INITIALIZE
C
C     GET FUNCTION KEY FROM MENU SCREEN
C
      CALL GETFKEY(IFKEY,1,4,ESC)
      CALL TCSSCR (10,0,10,79,0,7)
      IF(ESC) GO TO 100
      IF(IFKEY.EQ.1) ISAVE=0
      GO TO (10,20,30,40), IFKEY
C
C       HERE TO EDIT MODEL
C
10    CONTINUE
      AATEST=1.
      CALL TCSTXT (4, 0,0,14,36,'F1  Modify System Model              ')
      CALL TCSSCR (10,0,23,79,0,7)

      CALL EDIT
      GO TO 1
C
20    CONTINUE
      CALL TCSSCR (10,0,23,79,0,7)
      CALL TCSTXT (5, 0,0,14,36,'F2  File Operations - Store/Retreive')
      CALL TCSTXT (10,0,0,11,26,'MODEL FILE OPERATIONS MENU')
      CALL TCSTXT (11,0,0, 9,23,'F1  Store Model in File')
      CALL TCSTXT (12,0,0, 9,28,'F2  Retrieve Model from File')
      CALL TCSTXT (13,0,0, 9,27,'F3  Show Directory of Files')
      CALL TCSTXT (15,0,0,11,71,'Tap F1, F2, or F3 to Indicate Your Choi
     1ce Or ESC To Go To The Main Menu')
4     CALL GETFKEY(IFKEY,1,3,ESC)
      IF(ESC) GO TO 1
      GO TO (81,82,83), IFKEY
      IF(IFKEY.EQ.1.OR.IFKEY.EQ.2) ISAVE=1
      GO TO 4
C
81    CONTINUE
      CALL TCSTXT (11,0,0,14,23,'F1  Store Model in File')
      CALL TCSSCR (20,0,23,79,0,7)
C
C     FILE MODEL
C
C    CHECK THAT NO MODEL EXISTS
```

```
C
      CALL NOMODEL(IER)
      IF(IER.NE.0) GO TO 1
      CALL SAVMODEL(0,BUFF16)
      GO TO 1
C
82    CONTINUE
      CALL TCSTXT (12,0,0,14,28,'F2  Retrieve Model from File')
      CALL TCSSCR (20,0,23,79,0,7)
C
C     GET MODEL
C
      CALL GETMODEL(0,BUFF16)
      GO TO 1
C
83    CONTINUE
C
C     SHOW DIRECTORY
C
C    OPEN DIRECTORY WINDOW IN UPPER RIGHT HAND CORNER
C
      CALL TCSSCR (13,0,21,79,0,7)
      CALL SHODIR(0)
      NOBUFF=0
C
      CALL TCSTXT (11,0,0, 9,23,'F1  Store Model in File')
      CALL TCSTXT (12,0,0, 9,28,'F2  Retrieve Model from File')
      CALL TCSTXT (12,0,0, 0,36,'                                    ')
      CALL TCSTXT (20,0,0,11,66,'Tap F1 or F2 to Indicate Your Choice Or
     1 ESC To Go To The Main Menu')
240   CALL TCSCUR(0,6,7)
      CALL TCSKEY(ISCAN,IASC,NCHAR)
      IF(IASC.EQ.27.AND.NCHAR.EQ.1) GO TO 1
C
      IF(NCHAR.EQ.1.AND.(IASC.LT.32.OR.IASC.GT.122).AND.IASC.NE.8
     1     .AND.IASC.NE.13) THEN
                              CALL TCSBEL
                              CALL TCSBEL
                              GO TO 240
                              ENDIF
C
C  CHECK TO SEE IF CURSOR KEY DEPRESSED
C
C  MOVE CURSOR UP
C
      IF(NCHAR.EQ.2.AND.ISCAN.EQ.72) CALL SHODIR(1)
C
C   MOVE CURSOR DOWN
C
      IF(NCHAR.EQ.2.AND.ISCAN.EQ.80) CALL SHODIR(2)
```

```
C
C    KEY IS NOT CURSOR KEY, CONTINUE TO HANDLE MENU CHOICE
C
243    IF(NCHAR.EQ.1) GO TO 240
       IF(NCHAR.EQ.2.AND.(ISCAN.EQ.72.OR.ISCAN.EQ.80)) GO TO 240
       IFKEY=ISCAN-58
       GO TO (231,232), IFKEY
       CALL TCSBEL
       GO TO 240
C
231    CALL TCSSCR(14,0,14,79,0,7)
       CALL TCSTXT (10,0,0,14,23,'F1  Store Model in File')
       KFLAGF=1
       CALL TCSCUR(1,6,7)
       CALL TCSSCR(14,0,14,79,0,7)
       CALL TCSTXT (14,0,0,11,27,'Enter The Output Filename :')
       CALL TCSTXT (14,29,0,31,26,'                          ')
       CALL TCSTXT (14,30,0,31,1,' ')
       GO TO 233
C
232    CALL TCSSCR(14,0,14,79,0,7)
       CALL TCSTXT (11,0,0,14,28,'F2  Retrieve Model from File')
       KFLAGF=2
       CALL TCSCUR(1,6,7)
       CALL TCSSCR(14,0,14,79,0,7)
       CALL TCSTXT (14,0,0,11,26,'Enter The Input Filename :')
       CALL TCSTXT (14,28,0,31,26,'                          ')
       CALL TCSTXT (14,30,0,31,1,' ')
C
233    ICOLNO=30
234    CONTINUE
C
235    CONTINUE
       CALL TCSCUR(1,6,7)
       IF(NOBUFF.NE.0) CALL TCSTXT(14,ICOLNO,0,31,1,BUFF(NOBUFF))
       IF(NOBUFF.EQ.0) CALL TCSTXT(14,30,0,31,1,' ')
       CALL TCSKEY(ISCAN,IASC,NCHAR)
       IF(IASC.EQ.27.AND.NCHAR.EQ.1) GO TO 1
       CALL TCSCUR(0,6,7)
       IF(NCHAR.EQ.1.AND.(IASC.LT.32.OR.IASC.GT.122).AND.IASC.NE.8
     1      .AND.IASC.NE.13) THEN
                              CALL TCSBEL
                              GO TO 235
                              ENDIF
C
C  CHECK TO SEE IF CURSOR KEY DEPRESSED
C
C  MOVE CURSOR UP
C
       IF(NCHAR.EQ.2.AND.ISCAN.EQ.72) CALL SHODIR(1)
```

```
C
C   MOVE CURSOR DOWN
C
     IF(NCHAR.EQ.2.AND.ISCAN.EQ.80) CALL SHODIR(2)
C
C    KEY IS NOT CURSOR KEY, CONTINUE TO HANDLE BUFFERING
C
     IF(NCHAR.EQ.2.AND.(ISCAN.EQ.72.OR.ISCAN.EQ.80)) GO TO 235
C
250  CONTINUE
C
C    FIRST ADD KEY TO BUFFER AND SHOW ON SCREEN
C
     IF(NCHAR.EQ.1.AND.(IASC.NE.8.AND.IASC.NE.13)) THEN
                      NOBUFF=NOBUFF+1
                      IF(NOBUFF.GT.10) NOBUFF=10
                      BUFF(NOBUFF)=CHAR(IASC)
                      ICOLNO=30+NOBUFF
                      CALL TCSTXT(14,ICOLNO,0,31,1,BUFF(NOBUFF))
                      GO TO 235
                      ENDIF
C
C  HANDLE BS (CHARACTER DELETE)
C
     IF(NCHAR.EQ.1.AND.IASC.EQ.8)   THEN
C                     CALL TCSCUR(1,6,7)
                      IF(NOBUFF.EQ.0) THEN
                                 CALL TCSBEL
                                 GO TO 235
                                 ENDIF
                      ICOLNO=30+NOBUFF
                      CALL TCSTXT(14,ICOLNO,0,31,1,' ')
                      IF(NOBUFF.GT.0) THEN
                                 BUFF(NOBUFF)=' '
                                 NOBUFF=NOBUFF-1
                                 ENDIF
                      NCOLNO=30+NOBUFF
                      IF(NOBUFF.GT.0)
    1                      CALL TCSTXT(14,NCOLNO,0,31,1,BUFF(NOBUFF))
                      GO TO 235
                      ENDIF
C
C    HERE HANDLE CR TO TERMINATE BUFFER, READ AND GO
C
     IF(NOBUFF.EQ.0) GO TO 235
     IF(NCHAR.EQ.1.AND.IASC.EQ.13) THEN
                         IF(NOBUFF.EQ.0) GO TO 235
                         MBUFF=NOBUFF+1
                         DO 281 LLU=MBUFF,80
281                      BUFF(LLU)=' '
```

```
                          NOBUFF=0
                          WRITE(LBUFF,282) BUFF
282                       FORMAT(80A1)
C
C     READ FILE NAME
C
                          READ(LBUFF,283) BUFF16
283                       FORMAT(A16)
                          IF(KFLAGF.EQ.1) CALL SAVMODEL(1,BUFF16)
                          IF(KFLAGF.EQ.2) CALL GETMODEL(1,BUFF16)
                          GO TO 1
                          ENDIF
      GO TO 235
C
30    CONTINUE
C
      IF (IFKEY .NE. 3) GO TO 32
C
C     REVIEW DESIGN INDEP/DEPENDENT PARAMETERS
C
      CALL TCSTXT (6,0,0,14,38,'F3  Review Design Indep/Dep Parameters')
      CALL TCSSCR (0,0,23,79,0,7)
      CALL REVIEW
32    CONTINUE
      GO TO 1
C
40    CONTINUE
      AATEST=4.
C
C     PERFORM SENSITIVITY ANALYSIS
C
      CALL TCSTXT (7,0,0,14,38,'F4  Perform Sensitivity Analysis')
      CALL TCSSCR (0,0,23,79,0,7)
      CALL SENSITIVE
      GO TO 1
C
100   CONTINUE
C
C     QUIT
C
      IF(ISAVE.EQ.1) THEN
                     CALL TCSMOD(0)
                     STOP ' RE FINISHED'
                     ENDIF
C
C    QUIT - END PROGRAM
C
      CALL TCSTXT (8,0,0,14,8,'ESC Exit')
      CALL TCSSCR (13,0,23,79,0,7)
      CALL TCSCUR(0,6,7)
```

```
C
C  CHECK TO SEE IF MODEL EXISTS, IF NOT THEN QUIT
C
      IF(DATA(2).NE.0..AND.DATA(4).NE.0..AND.DATA(5).NE.0..AND.
     1   DATA(6).NE.0..AND.DATA(7).NE.0..AND.DATA(8).NE.0..AND.
     1   DATA(9).NE.0..AND.DATA(10).NE.0..AND.DATA(11).NE.0..AND.
     1   DATA(12).NE.0..AND.DATA(13).NE.0..AND.DATA(14).NE.0..AND.
     1   DATA(15).NE.0..AND.DATA(16).NE.0..AND.DATA(17).NE.0..AND.
     1   DATA(18).NE.0..AND.DATA(19).NE.0..AND.DATA(20).NE.0..AND.
     2   DATA(21).NE.0.) GO TO 102
      CALL TCSMOD(0)
      STOP
C
102   CALL TCSTXT (15,0,0,11,31,'Current Model Not Saved To Disk')
      CALL TCSTXT (17,0,0, 9,28,'   F1  To Save Model To Disk')
      CALL TCSTXT (18,0,0, 9,35,'   F2  To Exit Without Saving Model')
      CALL TCSTXT(20,0,0,11,54,'Tap A Key For Your Choice - Esc To Retur
     1n To Main Menu')
C
C    GET FUNCTION KEY FROM MENU SCREEN
C
      CALL GETFKEY(IFKEY,1,2,ESC)
C
      IF(ESC) GO TO 1
      IF(IFKEY.EQ.2) THEN
                     CALL TCSMOD(0)
                     STOP ' RE FINISHED'
                     ENDIF
      CALL TCSTXT (10,0,0,9,11,'ESC Exit RE')
      CALL TCSSCR (12,0,23,79,0,7)
      CALL TCSTXT (13,0,0,11,34,'Tap A Function Key For Your Choice')
      ISAVE=1
      GO TO 20
      END
C
      SUBROUTINE LOGO
C
C   PUTS UP ALL STARTING SCREENS
C
      CALL TCSBLD('RELOGO.PIC ',0,0,1)
      CALL TCSKEY(ISCAN,IASC,NCHAR)
      CALL TCSMOD(0)
      RETURN
      END
C
      SUBROUTINE GETMODEL(ISKIP,BUFF16)
C
C     READS MODEL FILE
C
      CHARACTER*16 BUFF16
```

```
      CHARACTER*20 IFILE
      CHARACTER*50 FILE50
      REAL*4 DATA
      LOGICAL FEXST
      COMMON /CINP/ DATA(21)
      IF(ISKIP.EQ.1) THEN
                     READ(BUFF16,1) IFILE
1                    FORMAT(A16,4X)
                     GO TO 2
                     ENDIF
C
C    OPEN THE DISK FILE FOR INPUT.
C
      CALL TCSSCR(20,0,20,79,0,7)
      CALL TCSCUR(1,6,7)
      CALL TCSTXT (21,0,0,11,26,'Enter The Input Filename :')
      CALL TCSTXT (21,28,0,31,26,'                              ')
      CALL TCSTXT (21,29,0,31,1,' ')
97    READ(*,301,END=99,ERR=99) IFILE
301   FORMAT(A)
      CALL TCSCUR(0,6,7)
2     WRITE(FILE50,21) IFILE
21    FORMAT(A)
      GO TO 98
99    CALL TCSBEL
      CALL TCSBEL
      CALL TCSBEL
      GO TO 97
C
98    INQUIRE(FILE=FILE50,EXIST=FEXST,IOSTAT=IER)
      IF(.NOT.FEXST) GO TO 501
      OPEN(UNIT=4,FILE=FILE50,STATUS='OLD',IOSTAT=IER)
      IF(IER.NE.0) GO TO 503
C
C     CLEAN OUT MODEL STORAGE
C
      CALL PURGMOD
      READ(4,*,END=96,ERR=96) DATA
      CLOSE(UNIT=4)
      RETURN
C
96         CALL TCSMOD(0)
           CALL PURGMOD
           CALL TCSCUR(0,6,7)
           CALL TCSTXT( 8,24,0,4,31,'VDDDDDDDDDDDDDDDDDDDDDDDDDDDDD7')
           CALL TCSTXT( 9,24,0,4,31,':  DEFECTIVE MODEL READ FROM  :')
           CALL TCSTXT(10,24,0,4,31,':             DISK            :')
           CALL TCSTXT(11,24,0,4,31,'GDDDDDDDDDDDDDDDDDDDDDDDDDDDDD6')
           CALL TCSTXT(12,24,0,4,31,':    USE EDITOR TO REPAIR     :')
           CALL TCSTXT(13,24,0,4,31,'GDDDDDDDDDDDDDDDDDDDDDDDDDDDDD6')
```

```
          CALL TCSTXT(14,24,0,4,31,':   RETURNING TO MAIN MENU    :')
          CALL TCSTXT(15,24,0,4,31,'SDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD=')
          CALL TCSBEL
          CALL TCSBEL
          CALL TCSBEL
          CALL WAITTO
          CALL WAITTO
          CALL TCSMOD(0)
          RETURN
C
C   COME HERE IF FILE NOT FOUND OR DEFECTIVE FILE NAME
C
501   CONTINUE
      CALL TCSTXT(21,28,0,4,31,'VDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD7')
      CALL TCSTXT(22,28,0,4,31,':  FILE OR DEVICE NOT FOUND    :')
      CALL TCSTXT(23,28,0,4,31,':  CHECK DEFAULT PATH OR DIR   :')
      CALL TCSTXT(24,28,0,4,31,'SDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD=')
      CALL TCSBEL
      CALL TCSBEL
      CALL TCSBEL
      CALL WAITTNO
      CALL TCSMOD(0)
      RETURN
C
503   CONTINUE
      CALL TCSTXT(21,28,0,4,31,'VDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD7')
      CALL TCSTXT(22,28,0,4,31,':     IMPROPER FILE NAME       :')
      CALL TCSTXT(23,28,0,4,31,':   RETURNING TO MAIN MENU     :')
      CALL TCSTXT(24,28,0,4,31,'SDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD=')
      CALL TCSBEL
      CALL TCSBEL
      CALL TCSBEL
      CALL WAITTNO
      CALL TCSMOD(0)
      RETURN
      END
C
      SUBROUTINE SAVMODEL(ISKIP,BUFF16)
C
      LOGICAL RUTH,FEXST
      CHARACTER*16 BUFF16
      CHARACTER*1 FILE1(50),FILE2(50)
      CHARACTER*20 IOFILE,IBLANK
      CHARACTER*50 FILE50
      REAL*4 DATA
      COMMON /CINP/ DATA(21)
      DATA IBLANK / '                    '/
C
      IF(ISKIP.EQ.1) THEN
                      READ(BUFF16,1) IOFILE
```

```
1                    FORMAT(A16,4X)
                     RUTH=.FALSE.
                     GO TO 602
                     ENDIF
      CALL TCSCUR(1,6,7)
23    CALL TCSTXT (21,0,0,9,28,'Enter The Output Filename : ')
      CALL TCSTXT (21,29,0,31,28,'                            ')
      CALL TCSTXT (21,31,0,31,1,' ')
      READ(*,600,END=601,ERR=601) IOFILE
C
600   FORMAT(A)
      RUTH=IOFILE.EQ.IBLANK
      IF(IOFILE.EQ.IBLANK) GO TO 601
      GO TO 602
601   IOFILE='OUTPUT.PND'
      RUTH=.TRUE.
602   WRITE(FILE50,21) IOFILE
21    FORMAT(A20)
      CALL TCSCUR(0,6,7)
      INQUIRE(FILE=FILE50,EXIST=FEXST,IOSTAT=IER)
      IF(IER.NE.0) THEN
             CALL TCSTXT(22,28,0,4,31,'VDDDDDDDDDDDDDDDDDDDDDDDDDDDDD7')
             CALL TCSTXT(23,28,0,4,31,': ERROR IN FILE NAME, REENTER :')
             CALL TCSTXT(24,28,0,4,31,'SDDDDDDDDDDDDDDDDDDDDDDDDDDDDD=')
             CALL TCSBEL
             CALL TCSBEL
             CALL TCSBEL
             CALL WAITTO
             CALL TCSSCR(21,2,23,78,0,7)
             GO TO 23
             ENDIF
      OPEN(UNIT=4,FILE=FILE50,STATUS='NEW',IOSTAT=IER)
      IF(IER.NE.0) THEN
             CALL TCSTXT(22,28,0,4,31,'VDDDDDDDDDDDDDDDDDDDDDDDDDDDDD7')
             CALL TCSTXT(23,28,0,4,31,': ERROR IN FILE NAME, REENTER :')
             CALL TCSTXT(24,28,0,4,31,'SDDDDDDDDDDDDDDDDDDDDDDDDDDDDD=')
             CALL TCSBEL
             CALL TCSBEL
             CALL TCSBEL
             CALL WAITTO
             CALL TCSSCR(21,2,23,78,0,7)
             GO TO 23
             ENDIF
C
C    WRITE MODEL DATA
C
40    WRITE(4,*) DATA
171   CLOSE(UNIT=4)
      IF(RUTH) CALL TCSTXT(22,13,0,11,28,'Output Written to OUTPUT.PND')
      IF(.NOT.RUTH) CALL TCSTXT(22,13,0,11,18,'Output Written to ')
```

```
      IF(.NOT.RUTH) CALL TCSTXT(22,32,0,11,18,IOFILE)
      CALL WAITTO
      RETURN
      END
C
      SUBROUTINE PURGMOD
C
C     PURGES MODEL FROM COMMON
C
      REAL*4 DATA
      COMMON /CINP/ DATA(21)
      DO 505 N=1,21
505   DATA(N)=0.
      RETURN
      END
C
      SUBROUTINE EDIT
      REAL*4 DATA,XXX
      COMMON /CINP/ DATA(21)
      COMMON AATEST
      CHARACTER*33 LABEL33(21)
      CHARACTER*80 LBUFF
      CHARACTER*9 DAY9,DATE9
      CHARACTER*8 TIME8
      CHARACTER*1 BUFF(80)
      LOGICAL BS,CR,CURS
      CHARACTER*19 LABEL19(13)
C
      DATA LABEL19/
     1 'Evap Heat Rate, B/m',
     2 'Cond Heat Rate, B/m',
     3 '     No of Evap Fins',
     4 '     No of Cond Fins',
     5 '     Length Evap, In',
     6 '     Length Cond, In',
     7 'Compressor Motor Hp',
     8 '  Evap Fan Motor Hp',
     9 '  Cond Fan Motor Hp',
     * ' Acquisition Cost $',
     1 '     Design Life, Hr',
     2 '            MTTR, Hr',
     3 '   Overall MTBF, Hr'/
C
      DATA LABEL33/'              Number of Cylinders',
     1 '                Crank Speed, RPM','              Piston Diamete
     2r, In','  Maximum Cylinder Volume, Cu In',
     3 ' Cylinder Clearance Volume, Cu In',
     4 'Condenser Mean Temperature, Deg F',
     5 'Condenser   Inlet Air Temp, Deg F',
     6 'Condenser    Exit Air Temp, Deg F',
```

```
     7 'Evaporator  Inlet Air Temp, Deg F',
     8 'Evaporator   Exit Air Temp, Deg F',
     9 'Condenser       Fin Width, In (C)',
     * 'Condenser   Fin Thickness, In (C)',
     1 'Condenser       Fin Pitch, In (C)',
     2 'Condenser     Tube Radius, In (C)',
     3 'Condenser  Air Velocity, fpm (C)',
     4 'Evaporator      Fin Width, In (E)',
     5 'Evaporator  Fin Thickness, In (E)',
     6 'Evaporator      Fin Pitch, In (E)',
     7 'Evaporator    Tube Radius, In (E)',
     8 'Evaporator  Air Velocity, fpm (E)',
     9 'Evaporator Mean Temperature,Deg F'/
C
C    1     NCY     NO OF CYLS
C    2     RPM     CRANK SPEED, RPM
C    3     DP      PISTON DIA, IN
C    4     VMAX    MAX VOL/CYL, CUBIC IN
C    5     VCL     CL VOL/CYL, CUBIC IN
C    6     T3      COND TEMP, DEG F
C    7     TCIN    COND INLET AIR TEMP, DEG F
C    8     TCOUT   COND EXH AIR TEMP, DEG F
C    9     TEIN    EVAP INLET AIR TEMP, DEG F
C   10     TEOUT   EVAP EXH AIR TEMP, DEG F
C   11     D       COND FIN WIDTH, IN
C   12     TH      COND FIN THICKNESS, IN
C   13     LPI     COND FIN PITCH, IN
C   14     R       COND TUBE RADIUS, IN
C   15     UH      COND AIR VELOCITY, IN/S
C   16     ED      EVAP FIN WIDTH,IN
C   17     ETH     EVAP FIN THICKNESS, IN
C   18     EL      EVAP FIN PITCH, IN
C   19     ER      EVAP TUBE RADIUS, IN
C   20     EUH     EVAP AIR VELOCITY, IN/S
C   21     T1      EVAP TEMP, DEG F
C
1     CONTINUE
C
      CALL TCSMOD(0)
      CALL TCSSCR(0,0,24,79,0,0)
C
C    PUT UP COLUMN HEADERS
C
C      CALL DAY(DAY9)
C      CALL DATE(DATE9)
C      CALL TIME(TIME8)
      CALL TCSTXT(0,50,0,14,9,DAY9)
      CALL TCSTXT(0,60,0,14,9,DATE9)
      CALL TCSTXT(0,70,0,14,8,TIME8)
      CALL TCSTXT(0,2,0,14,29,'Refrigeration System Designer')
```

```
      DO 101 N=0,79
C     CALL TCSTXT( 0,N,0, 9,1,'M')
      CALL TCSTXT( 1,N,0, 9,1,'M')
      IF(N.GT.34)  THEN
                   CALL TCSTXT(21,N,0, 9,1,'M')
                   ENDIF
      CALL TCSTXT(23,N,0, 9,1,'M')
      IF(N.GE.45) CALL TCSTXT(16,N,0, 9,1,'D')
      IF(N.GE.45) CALL TCSTXT(18,N,0, 9,1,'D')
101   CONTINUE
      DO 102 N=1,23
      CALL TCSTXT( N, 0,0,9,1,':')
      CALL TCSTXT( N,79,0,9,1,':')
      IF(N.GE.2.AND.N.LE.23) CALL TCSTXT( N,34,0,9,1,'3')
      IF(N.GE.2.AND.N.LE.22) THEN
                             CALL TCSTXT( N,45,0,9,1,'3')
                             IF(N.LT.17) CALL TCSTXT( N,68,0,9,1,'3')
                             NN=N-1
                             CALL TCSTXT(N,1,0,3,33,LABEL33(NN))
                             IF(NN.LE.13)
     1                       CALL TCSTXT(N,47,0,2,19,LABEL19(NN))
                             CALL TCSREAL(N,35,0,14,DATA(NN))
                             ENDIF
102   CONTINUE
C
      CALL TCSTXT(17,46,0,2,20,'Expected Cooling,B/m')
      CALL TCSTXT(19,46,0,2,22,'Expected Annual Equiv.')
      CALL TCSTXT(20,47,0,2,18,'Life Cycle Cost  $')
C
      CALL TCSTXT( 1,0,0, 9,1,'I')
C     CALL TCSTXT( 1,0,0, 9,1,'L')
      CALL TCSTXT(23,0,0, 9,1,'H')
C
      CALL TCSTXT( 1,79,0, 9,1,';')
C     CALL TCSTXT( 1,79,0, 9,1,'9')
      CALL TCSTXT(21,79,0, 9,1,'9')
      CALL TCSTXT(23,79,0, 9,1,'<')
C
      CALL TCSTXT( 1,34,0, 9,1,'Q')
      CALL TCSTXT(23,34,0, 9,1,'O')
C
      CALL TCSTXT( 1,45,0, 9,1,'Q')
      CALL TCSTXT(16,45,0, 9,1,'C')
      CALL TCSTXT(21,45,0, 9,1,'F')
      CALL TCSTXT(22,45,0, 9,1,'3')
      CALL TCSTXT(23,45,0, 9,1,'O')
C
      CALL TCSTXT( 1,34,0, 9,1,'Q')
      CALL TCSTXT( 1,45,0, 9,1,'Q')
      CALL TCSTXT( 1,68,0, 9,1,'Q')
```

```
      CALL TCSTXT(16,68,0, 9,1,'A')
C
      CALL TCSTXT(16,79,0, 9,1,'6')
      CALL TCSTXT(24, 2,0,79,10,' F1 = T-S ')
      CALL TCSTXT(24,39,0,79,10,' F2 = P-H ')
      CALL TCSTXT(24,67,0,79,10,' F3 = P-V ')
C
      CALL TCSBLK(1,0)
C
      CALL TCSTXT(22,47,0,14,10,'Enter Data')
      CALL TCSSCR(22,60,22,70,0,96)
      CALL TCSCUR(0,6,7)
C
C     IROWC - ROW NO FOR COMPUTATION - ARRAY ELEMENT #'S
C     IROWS - SCREEN ROW NO
C
      ICOLNO=62
      IROWS=2
      IROWC=1
C
C   SET SCREEN CURSOR TO DEFAULT TOP POSITION
C
      AATEST=1.
C
      CALL TCSREAL(2,35,0,224,DATA(1))
      CALL COMPUTE
      CALL TCSTXT(22,62,0,96,1,' ')
C
71    CONTINUE
      CALL TCSCUR(1,6,7)
      CALL TCSKEY(ISCAN,IASC,NCHAR)
      IF(IASC.EQ.27.AND.NCHAR.EQ.1) RETURN
C
C   IF F1 F2 OR F3 PRESSED, THEN PLOT
C
C               1 T-S
C               2 P-H
C               3 P-V
C
      IF(ISCAN.GE.59.AND.ISCAN.LE.61) THEN
                        IF(ISCAN.EQ.59) CALL PLOT(1)
                        IF(ISCAN.EQ.60) CALL PLOT(2)
                        IF(ISCAN.EQ.61) CALL PLOT(3)
                        GO TO 1
                        ENDIF
      CALL TCSCUR(0,6,7)
      IROWCO=IROWC
      IROWSO=IROWS
      IF(NCHAR.EQ.1.AND.(IASC.LT.32.OR.IASC.GT.122).AND.IASC.NE.8
     1     .AND.IASC.NE.13) THEN
```

```
                              CALL TCSBEL
                              GO TO 71
                              ENDIF
C
C  CHECK TO SEE IF CURSOR KEY DEPRESSED
C
C  MOVE CURSOR UP
C
      IF(NCHAR.EQ.2.AND.ISCAN.EQ.72) GO TO 73
C
C   MOVE CURSOR DOWN
C
      IF(NCHAR.EQ.2.AND.ISCAN.EQ.80) GO TO 72
C
C    KEY IS NOT CURSOR KEY, GO TO 76 TO HANDLE BUFFERING
C
      GO TO 76
C
C   MOVE DOWN
C
72    CONTINUE
      IF(IROWC.EQ.21) THEN
                      CALL TCSBEL
                      CALL TCSTXT(22,ICOLNO,0,96,1,' ')
                      GO TO 76
                      ENDIF
C
C   REMOVE OLD ANNOTATION
C
      CALL TCSCUR(0,6,7)
      XXX=DATA(IROWC)
      CALL TCSREAL(IROWS,35,0,14,XXX)
      IROWC=IROWC+1
      IROWS=IROWS+1
C
C  PUT UP NEW ANNOTATION
C
      XXX=DATA(IROWC)
      CALL TCSREAL(IROWS,35,0,224,XXX)
      CALL TCSTXT(22,ICOLNO,0,96,1,' ')
      GO TO 76
C
C   MOVE UP
C
73    CONTINUE
      IF(IROWC.EQ.1) THEN
                      CALL TCSBEL
                      CALL TCSTXT(22,ICOLNO,0,96,1,' ')
                      GO TO 76
                      ENDIF
```

```
C
C   REMOVE OLD ANNOTATION
C
      CALL TCSCUR(0,6,7)
      XXX=DATA(IROWC)
      CALL TCSREAL(IROWS,35,0,14,XXX)
      IROWC=IROWC-1
      IROWS=IROWS-1
C
C  PUT UP NEW ANNOTATION
C
      XXX=DATA(IROWC)
      CALL TCSREAL(IROWS,35,0,224,XXX)
      CALL TCSTXT(22,ICOLNO,0,96,1,' ')
      GO TO 76
C
76    CONTINUE
C
C     CALL TCSCUR(0,6,7)
      CURS=.FALSE.
      CR=.FALSE.
      BS=.FALSE.
      IF((ISCAN.EQ.72.OR.ISCAN.EQ.80).AND.NCHAR.EQ.2) CURS=.TRUE.
      IF(IASC.EQ.13.AND.NCHAR.EQ.1) CR=.TRUE.
      IF(IASC.EQ.8.AND.NCHAR.EQ.1) BS=.TRUE.
C
C     GOOD KEYS AFTER HERE
C
C    FIRST ADD KEY TO BUFFER AND SHOW ON SCREEN
C
      IF((.NOT.CURS).AND.(.NOT.CR).AND.(.NOT.BS)) THEN
                      NOBUFF=NOBUFF+1
                      IF(NOBUFF.GT.9) NOBUFF=9
                      BUFF(NOBUFF)=CHAR(IASC)
                      ICOLNO=62+NOBUFF
                      CALL TCSTXT(22,ICOLNO,0,96,1,BUFF(NOBUFF))
                      GO TO 71
                      ENDIF
C
C  HANDLE BS (CHARACTER DELETE)
C
      IF(BS)          THEN
C                     CALL TCSCUR(1,6,7)
                      IF(NOBUFF.EQ.0) THEN
                                      CALL TCSBEL
                                      GO TO 71
                                      ENDIF
                      ICOLNO=62+NOBUFF
                      CALL TCSTXT(22,ICOLNO,0,96,1,' ')
                      IF(NOBUFF.GT.0) THEN
```

```
                                      BUFF(NOBUFF)=' '
                                      NOBUFF=NOBUFF-1
                                      ENDIF
                        NCOLNO=62+NOBUFF
                        IF(NOBUFF.GT.0)
     1                       CALL TCSTXT(22,NCOLNO,0,96,1,BUFF(NOBUFF))
                        GO TO 71
                        ENDIF
C
C     HERE HANDLE CURSOR KEYS, OR CR TO TERMINATE BUFFER, READ AND GO
C
      IF(NOBUFF.EQ.0) GO TO 71
      IF(CURS.OR.CR) THEN
                           IF(NOBUFF.EQ.0) GO TO 71
                           MBUFF=NOBUFF+1
                           DO 107 LLU=MBUFF,80
107                        BUFF(LLU)=' '
                           WRITE(LBUFF,108) BUFF
108                        FORMAT(80A1)
C
                           DO 111 LLQ=1,80
                           IF(BUFF(LLQ).NE.' ') GO TO 112
111                        CONTINUE
                           BUFF(2)='0'
                           WRITE(LBUFF,108) BUFF
112                        CONTINUE
                           NOBUFF=0
                           DO 997 LLP=1,80
997                        BUFF(LLP)=' '
                           CALL TCSSCR(22,60,22,70,0,96)
                           READ(LBUFF,*,END=110,ERR=110) XXX
                           GO TO 109
                           ENDIF
                  GO TO 71
C
C     INTERNAL READ ERROR HANDLER
C
110   CALL TCSTXT(22,61,0,207,6,' ERROR')
      XXX=0.
      NOBUFF=0
      DO 113 LLW=1,80
113   BUFF(LLW)=' '
      LBUFF='
     1                        '
      CALL TCSBEL
      CALL TCSBEL
      CALL TCSBEL
      CALL WAITTNO
      CALL TCSTXT(22,61,0,7,6,'      ')
C
```

```
      CALL TCSTXT(22,47,0,14,10,'Enter Data')
      CALL TCSSCR(22,60,22,70,0,96)
      CALL TCSCUR(0,6,7)
      CALL TCSTXT(22,62,0,96,1,' ')
C
      GO TO 71
C
109   CONTINUE
      IF(CR) THEN
            CALL TCSREAL(IROWS,35,0,224,XXX)
            DATA(IROWC)=XXX
            CALL TCSTXT(22,62,0,96,1,' ')
            ENDIF
      IF(CURS) THEN
            CALL TCSREAL(IROWSO,35,0,14,XXX)
            DATA(IROWCO)=XXX
            CALL TCSTXT(22,62,0,96,1,' ')
            ENDIF
      AATEST=1.
C       ******INPUT ERROR TRAPPING*****
      IF(DATA(1).LT.1..OR.(DATA(2).NE.3600..AND.DATA(2).NE.1800..AND.
     $DATA(2).NE.1200..AND.DATA(2).NE.900.).OR.
     $DATA(3).LT.1..OR.DATA(4).LT.1.5.OR.DATA(5).LT..1.OR.DATA(6).LE.
     $DATA(21).OR.DATA(7).GE.DATA(8).OR.DATA(7).LT.-457..OR.DATA(9)
     $.LE.DATA(10).OR.DATA(10).LT.-457..OR.DATA(11).LE.(2.*DATA(14))
     $.OR.DATA(14).LT..125.OR.DATA(12).LT..02.OR.DATA(13).LT.
     $(DATA(12)+.1).OR.DATA(15).LE.0..OR.DATA(16).LE.(2.*DATA(19))
     $.OR.DATA(19).LT..125.OR.DATA(17).LT..02.OR.DATA(18).LT.
     $(DATA(17)+.1).OR.DATA(20).LE.0..OR.DATA(21).LT.-457.)
     $GO TO 110
C
C    HAS DATA(1) BEEN RE-ENTERED AS A REAL #? CHANGE TO INTEGER
C     DATA(1)=INT(DATA(1))
C    PAS2 COMPILER ERROR?
      CALL COMPUTE
      CALL TCSCUR(0,6,7)
      CALL TCSTXT(22,62,0,96,1,' ')
      GO TO 71
C
3     CONTINUE
      RETURN
      END
C
      SUBROUTINE SHODIR(IARG)
C
C     SHOW DIRECTORY
C
C     IARG  =  0   INITIALIZE, GET AND PUT UP DIR
C              1   SCROLL UP
C              2   SCROLL DOWN
```

```
C
      CHARACTER*16 DIRENT(100)
      CHARACTER*1 LNAME(72),MNAME(72)
      CHARACTER*72 NNAME,KNAME
      DATA ITOP,IBOT/0,11/
C
      IF(IARG.EQ.1) GO TO 241
      IF(IARG.EQ.2) GO TO 242
      IF(IARG.NE.0) RETURN
C
C     SHOW DIRECTORY
C
C    OPEN DIRECTORY WINDOW IN UPPER RIGHT HAND CORNER
C
      CALL TCSTXT(3,57,0,12,9,'DIRECTORY')
C     CALL TCSTXT(5,48,0,3,28,DRVPTH)
      DO 201 N=46,78
      CALL TCSTXT( 2,N,0,11,1,'M')
      CALL TCSTXT( 4,N,0,11,1,'M')
      CALL TCSTXT( 6,N,0,11,1,'M')
      CALL TCSTXT(17,N,0,11,1,'M')
      CALL TCSTXT(19,N,0,11,1,'M')
201   CONTINUE
      DO 202 N=2,19
      CALL TCSTXT( N,46,0,11,1,':')
202   CALL TCSTXT( N,78,0,11,1,':')
C
      CALL TCSTXT( 2,46,0,11,1,'I')
      CALL TCSTXT( 4,46,0,11,1,'L')
      CALL TCSTXT( 6,46,0,11,1,'L')
      CALL TCSTXT(17,46,0,11,1,'L')
      CALL TCSTXT(19,46,0,11,1,'H')
      CALL TCSTXT( 2,78,0,11,1,';')
      CALL TCSTXT( 4,78,0,11,1,'9')
      CALL TCSTXT( 6,78,0,11,1,'9')
      CALL TCSTXT(17,78,0,11,1,'9')
      CALL TCSTXT(19,78,0,11,1,'<')
C
      CALL TCSTXT(18,50,0, 7,25,'Use  To Scroll Directory')
C
C       PUT DEFAULT DRIVE/PATH INTO CHARACTER VARIABLE
C
      WRITE(KNAME,206)
206   FORMAT('PENDDIR.TMP ')
      WRITE(NNAME,205)
205   FORMAT('*.*',69X)
C
C   PUT DIR INTO FILE, COUNT # OF ENTRIES, AND OPEN FILE
C
      CALL TCSDIR(0,NNAME,NENT,2,KNAME)
```

```
C
      OPEN(UNIT=87,FILE=KNAME,STATUS='OLD')
C
      IF(NENT.GT.99) NENT=99
      DO 208 KY=1,NENT
207   READ(87,209,END=210,ERR=210) DIRENT(KY)
209   FORMAT(A16)
      IF(DIRENT(KY).EQ.'PENDDIR.TMP      ') THEN
                                              NENT=NENT-1
                                              GO TO 207
                                              ENDIF
208   CONTINUE
C210   CLOSE(UNIT=87,STATUS='DELETE')
210   CLOSE(UNIT=87)
C
      DO 211 N=1,10
      IROW=N+6
      CALL TCSTXT(IROW,52,0,2,16,DIRENT(N))
211   CONTINUE
C
      RETURN
C
C   MOVE DOWN
C
241   CONTINUE
      IF(ITOP.GT.0) GO TO 244
      CALL TCSCUR(0,6,7)
      CALL TCSBEL
      RETURN
244   CALL TCSSCR(7,47,16,77,-1,2)
      CALL TCSTXT(7,52,0,2,16,DIRENT(ITOP))
      CALL TCSCUR(0,6,7)
      ITOP=ITOP-1
      IBOT=IBOT-1
      RETURN
C
C   MOVE UP
C
242   CONTINUE
      IF(ITOP.LT.(NENT-10)) GO TO 245
      CALL TCSBEL
      CALL TCSCUR(0,6,7)
      RETURN
245   CONTINUE
      CALL TCSSCR(7,47,16,77,1,2)
      CALL TCSTXT(16,52,0,2,16,DIRENT(IBOT))
      CALL TCSCUR(0,6,7)
      ITOP=ITOP+1
      IBOT=IBOT+1
      RETURN
```

```
C
      END
C
        SUBROUTINE BSAVE
        CHARACTER*1 BUFF(72)
        CHARACTER*72 NNAME
        DIMENSION ISTR(23)
        DATA ISTR/69,78,84,69,82,32,79,85,84,80,85,84,32,70,73,76,69,
     1  78,65,77,69,58,32/
C
C                 E  N  T  E  R    O  U  T  P  U  T    F  I  L  E
C        N  A  M  E  :
C
C     GOT ALT-H,  NOW BSAVE TO DISK
C
        CALL MOVABS(0,0)
        CALL ANSTR(23,ISTR)
        DO 30 I=1,72
30      BUFF(I)=' '
        CALL GET2BUF(BUFF,72)
C
C    EXTRACT 1ST 20 NONBLANK CHARS IN BUFF AND PUT INTO FNAME
C
        WRITE(NNAME,15) BUFF
15      FORMAT(72A1)
        IX=0
        IY=0
        JBEAMX=IFIX(FLOAT(IX)*.624+.50)
        LBEAMX=IFIX(FLOAT(1021+IX)*.624+.50)
        IY=IY-1
        DO 10 I=1,26
        IY=IY+1
        JBEAMY=199-IFIX(FLOAT(IY)*.255+.50)
10      CALL TCSLIN(JBEAMX,JBEAMY,LBEAMX,JBEAMY,0)
        CALL TCSBSV(NNAME,IERR)
        IF(IERR.NE.0) THEN
          CALL TCSBEL
          CALL TCSBEL
          CALL TCSBEL
          CALL STROUT(0,0,'ERROR IN WRITING FILE - DISK FULL',33)
          CALL WAITTNO
          CALL WAITTNO
          IX=0
          IY=0
          JBEAMX=IFIX(FLOAT(IX)*.624+.50)
          LBEAMX=IFIX(FLOAT(1021+IX)*.624+.50)
          IY=IY-1
          DO 40 I=1,26
          IY=IY+1
          JBEAMY=199-IFIX(FLOAT(IY)*.255+.50)
```

```
40        CALL TCSLIN(JBEAMX,JBEAMY,LBEAMX,JBEAMY,0)
          ENDIF
        RETURN
        END
C
        SUBROUTINE GET2BUF(BUFR,MAXSIZE)
        CHARACTER*1 BUFR(1)
        INTEGER BUFF(72)
        CHARACTER*1 BLK
        DATA BLK/' '/
        J=0
        DO 40 I=1,72
41      CALL TCSKEY(NSCAN,IASC,NCH)
C
C   HANDLE C/R, JUMP OUT OF INPUT LOOP
C
        IF(IASC.EQ.13.AND.NCH.EQ.1) GO TO 50
C
C   TRAP OUT ON F KEY AND RING BELL
C
        IF(NCH.NE.2) GO TO 43
           CALL BELL
           CALL BELL
           CALL BELL
           GO TO 41
C
C    PROCESS NON-F KEY, VALID INPUT
C
43      CONTINUE
C
C    HANDLE BACK SPACE
C
        IF(IASC.EQ.8) THEN
             J=J-1
             CALL BAKSP
             CALL CHRERAS
             GO TO 41
        ENDIF
C
C    INCREMENT BUFFER COUNTER
C
        J=J+1
C
C   MAP LC BACK TO UC
C
        IF(IASC.GE.97.AND.IASC.LE.122) IASC=IASC-32
C
C    WRITE TO SCREEN AND BUFFER CHARACTER
C
        CALL ANCHO(IASC)
```

```
        BUFF(J)=IASC
C
40      CONTINUE
C
50      NCHAR=J
        IF(NCHAR.LT.0) NCHAR=0
C
C   CLEAN OUT UPPER END OF BUFFER
C
        NN=NCHAR+1
        IF(NCHAR.EQ.0) NN=1
        DO 35 I=NN,72
35      BUFF(I)=32
C
C   LOAD BUFR WITH CHARACTERS
C
        DO 30 I=1,MAXSIZE
        BUFR(I)=CHAR(BUFF(I))
30      CONTINUE
        RETURN
        END
C
        SUBROUTINE CHRERAS
        COMMON /TKTRNX/ TMINVX,TMINVY,TMAXVX,TMAXVY,TREALX,TREALY,
     +  TIMAGX,TIMAGY,TRCOSF,TRSINF,TRSCAL,TRFACX,TRFACY,
     +  TRPAR1,TRPAR2,TRPAR3,TRPAR4,TRPAR5,TRPAR6,KMOFLG(2),
     +  KGNMOD,KPADV,KACHAR,KOBLEN,KTRAIL,KLEVEL,KPAD2,
     +  KCOLOR,KGNFLG,KGRAFL,KHOMEY,KKMODE,KHORSZ,KVERSZ,KTBLSZ,
     +  KSIZEF,KLMRGN,KRMRGN,KFACTR,KTERM,KLINE,KZAXIS,KBEAMX,KBEAMY,
     +  KMOVEF,KPCHAR(5),KDASHT,KMINSX,KMINSY,KMAXSX,KMAXSY,KEYCON,
     +  KINLFT,KOTLFT,KUNIT
        IX=KBEAMX
        IY=KBEAMY-24
        JBEAMX=IFIX(FLOAT(IX)*.624+.50)
        LBEAMX=IFIX(FLOAT(14+IX)*.624+.50)
        IY=IY-1
        DO 10 I=1,26
        IY=IY+1
        JBEAMY=199-IFIX(FLOAT(IY+14)*.255+.50)
  10    CALL TCSLIN(JBEAMX,JBEAMY,LBEAMX,JBEAMY,0)
        RETURN
        END
C
C-----SUBROUTINE--STROUT----
C
C PURPOSE:
C     THIS ROUTINE WRITES OUT A STRING OF CHARACTERS
C
      SUBROUTINE STROUT(IX,IY,ISTR,NO)
C
```

```
C     IX,IY    -  1024X780 COORDS OF STARTING PT OF STRING
C     ISTR     -  STRING OF CHARS, 4 CHARS PER WORD
C     NO       -  NO OF CHARS IN ISTR
C
      DIMENSION ISTR(1),JSTR(80),KSTR(80)
      CHARACTER*80 IBUF
      CHARACTER*4 ISTR
      CHARACTER*1 JSTR
C
C  START OF EXECUTABLE CODE
C
      CALL MOVABS(IX,IY)
      NX=(NO+3)/4
      WRITE(IBUF,10) (ISTR(LL),LL=1,NX)
10    FORMAT(20A4)
      READ(IBUF,20) JSTR
20    FORMAT(80A1)
      DO 40 J=1,NO
40    KSTR(J)=ICHAR(JSTR(J))
      CALL ANSTR(NO,KSTR)
      RETURN
      END
C
      SUBROUTINE TCSDLY(DELAY)
C
C   THIS SUBROUTINE WILL RUN IN AN IDLE LOOP FOR "DELAY" SECONDS
C   THEN RETURN TO THE CALLING ROUTINE.
C
      REAL*8 TST,TEND
      CALL TCSTIM(IDAY,IMO,IYR,IDW,IHR,IMIN,ISS,IHS)
      IS=IHR*3600+IMIN*60+ISS
C
      G=FLOAT(IYR)
      F=FLOAT(IMO+1)
      D=FLOAT(IDW)
      IF(IMO.LE.2) THEN
                   G=FLOAT(IYR-1)
                   F=FLOAT(IMO+13)
                   ENDIF
      TEND=DBLE(365.25*G+30.6*F+D-621049.)*86400.+DBLE(IS)
     1      +DBLE(IHS)/100.+DBLE(DELAY)
C
5     CALL TCSTIM(IDAY,IMO,IYR,IDW,IHR,IMIN,ISS,IHS)
      IS=IHR*3600+IMIN*60+ISS
C
      G=FLOAT(IYR)
      F=FLOAT(IMO+1)
      D=FLOAT(IDW)
      IF(IMO.LE.2) THEN
                   G=FLOAT(IYR-1)
```

```
                 F=FLOAT(IMO+13)
                 ENDIF
      TST=DBLE(365.25*G+30.6*F+D-621049.)*86400.+DBLE(IS)
     1      +DBLE(IHS)/100.
      IF(TST.GE.TEND) RETURN
      GO TO 5
      END
C
C
      SUBROUTINE COMPUTE
      COMMON TEMP(21)
      COMMON EAETSC
      COMMON AATEST
      COMMON BBTEST
      COMMON REFCST
      COMMON RFCOOL
C
C     SUBROUTINE FOR COMPUTING THE PERFORMANCE AND COST PARS FOR SYSTEM
C
      DIMENSION T(28), P(28), VG(28), HF(28), HG(28), SG(28), SF(28)
     $,PS(7), TS(9), VS(9,7), HS(9,7), SS(9,7), VP(9), HP(9), SP(9)
     $,TA(7),VT(7),KT(7),PRT(7),CPT(7),THP(39)
C
C     COMMON/CINP/T1,T3,NCY,VCL,VMAX,RPM,TCIN,TCOUT,TEIN,TEOUT,
C    $D,TH,R,LPI,UH,ED,ETH,ER,EL,EUH,DP,CCASU,CSTEU,CUAL
C
      COMMON/CINP/NCY,RPM,DP,VMAX,VCL,T3,TCIN,TCOUT,TEIN,TEOUT,
     $ D,TH,LPI,R,UH,ED,ETH,EL,ER,EUH,T1
C
C    1    NCY    NO OF CYLS
C    2    RPM    CRANK SPEED, RPM
C    3    DP     PISTON DIA, IN
C    4    VMAX   MAX VOL/CYL, CUBIC IN
C    5    VCL    CL VOL/CYL, CUBIC IN
C    6    T3     COND TEMP, DEG F
C    7    TCIN   COND INLET AIR TEMP, DEG F
C    8    TCOUT  COND EXH AIR TEMP, DEG F
C    9    TEIN   EVAP INLET AIR TEMP, DEG F
C   10    TEOUT  EVAP EXH AIR TEMP, DEG F
C   11    D      COND FIN WIDTH, IN
C   12    TH     COND FIN THICKNESS, IN
C   13    LPI    COND FIN PITCH, IN
C   14    R      COND TUBE RADIUS, IN
C   15    UH     COND AIR VELOCITY, IN/S
C   16    ED     EVAP FIN WIDTH,IN
C   17    ETH    EVAP FIN THICKNESS, IN
C   18    EL     EVAP FIN PITCH, IN
C   19    ER     EVAP TUBE RADIUS, IN
C   20    EUH    EVAP AIR VELOCITY, IN/S
C   21    T1     EVAP TEMP, DEG F
```

```
C
      COMMON/COUT/S1,P1,V1,H1,T2,S2,P2,V2,H2,S3,H3,H4,S4,WIN,QIN,
     $QOUT,COP,CR,MIN,WDIN,QDIN,QDOUT,MDC,MDE,N,W,NE,WE,WCAST,WSTEE,
     $CCOMM,THPCO,THPF,SG3, VA, VB, VC, VD,T4,P3,P4,CCM,CFM,COSTAL,WAL
C
      COMMON/REPS/IM,INN,JN
      COMMON/RESULTS/RESULTS(14)
C
      REAL NCY, MIN, MDC, MDE, NU,LPI,KCO,KT,MDCS,MDES
C
      COMMON/INDEP/ IDEMAND, XINT, SHORT, RFAC, RFAOP
C                                                                   ^
C     IDEMAND = DEMAND                                              ^
C     XINT = INTerst rate                                           ^
C     SHORT = SHORTage penalty.                                     ^
C     RFAC = annual Repair FACilty cost.                            ^
C     RFAOP = Repair Facilty Annual OPerating cost.                 ^
C                                                                  ^
      COMMON/DEP/ACQ,SALV,IDLIFE,AOPFU,AOPLA,AOPPM,AOPOTH,XMTBF(20)
     1 ,XMTTR(20),ES
C                                                                   ^
C     ACQ = ACQuisition cost.                                       ^
C     SALV = SALVage value at end of design life.                   ^
C     IDLIFE = Design LIFE of the equipment                         ^
C     AOPFU = Annual OPerating cost (FUel).                         ^
C     AOPLA = Annual OPerating cost (LAbor).                        ^
C     AOPPM = Annual OPerating cost (Preventive Maintenance).       ^
C     AOPOTH = Annual OPerating cost (OTHer).                       ^
C     XMTBF = array containing Mean Time Between Failures.          ^
C     XMTTR = array containing Mean Time To Repair the equipment.   ^
C                                                                   ^
C     MTTR and MTBF are defined as arrays of length 20.  If the      ^
C     design life of the eqiupment in question is greater than this  ^
C     (if IDLIFE > 20), these arrays must be increased in length.    ^
C                                                                    ^
C       **CHECK CALC. FOR IDLIFE TO SEE IF LIMITED TO 20***          ^
C
C     LIST FOR MOTORS WITH DIFFERENT HP
C
      DATA THP/.008,.01,.0125,.0143,.0167,.0182,.02,.025,.0286,.0333,
     $.04,.05,.0667,.0833,.1,.125,.167,.25,.333,.5,.75,1.,1.5,2.,3.,
     $5.,7.5,10.,15.,20.,25.,30.,40.,50.,60.,75.,100.,125.,150./
C
C     TEMP. TABLE FOR AIR (DEGREE F)
C
      DATA TA/40.,60.,80.,100.,120.,140.,160./
C
C     KINEMATIC VISCOSITY TABLE FOR AIR
C
      DATA VT/.5301,.5685,.6078,.6482,.6895,.7316,.7747/
```

```
C
C     THERMAL CONDUCTIVITY TABLE FOR AIR
C
      DATA KT/.01416,.01466,.01516,.01566,.01615,.01664,.01712/
C
C     PRANDTL TABLE FOR AIR
C
      DATA PRT/.714,.712,.709,.706,.703,.7,.698/
C
C     SPECIFIC HEAT TABLE FOR AIR
C
      DATA CPT/.2404,.2404,.2405,.2406,.2407,.2409,.2411/
C
C     PRESSURE TABLE FOR SUPERHEATED R-12
C     ONE DIMENSION
C
      DATA PS/100.,120.,140.,160.,180.,200.,300./
C
C     TEMP. TABLE FOR SUPERHEATED R-12
C     ONE DIMENSION
C
      DATA TS/100.,120.,140.,160.,180.,200.,250.,300.,350./
C
C     SPECIFIC VOLUME TABLE FOR SUPERHEATED R-12
C     TWO DIMENSION
C
      DATA VS/0.43138,0.45562,0.47881,0.50118,0.52291,0.54413,0.59549,
     $0.64518,1.,0.34655,0.36841,0.38901,0.4087,0.42766,0.44606,0.49025,
     $0.53267,1.,0.,0.30549,0.32445,0.34232,0.35939,0.37584,0.41499,
     $0.45226,1.,0.,0.25764,0.27558,0.29224,0.30797,0.32301,0.35847,
     $0.3919,1.,0.,0.,0.2371,0.25297,0.26775,0.28176,0.31142,0.34492,
     $0.37409,0.,0.,0.20579,0.22121,0.23535,0.2486,0.27911,0.3073,
     $0.33408,0.,0.,0.,0.,0.13482,0.14697,0.1723,0.19402,0.21388/
C
C     SPECIFIC ENTHALPY TABLE FOR SUPERHEATED R-12
C     TWO DIMENSION
C
      DATA HS/88.694,92.116,95.507,98.884,102.257,105.633,114.119,
     $122.707,1000.,87.675,91.237,94.736,98.199,101.642,105.076,113.67,
     $122.333,1000.,0.,90.297,93.923,97.483,101.003,104.501,113.212,
     $121.953,1000.,0.,89.283,93.059,96.732,100.34,103.907,112.743,
     $121.567,1000.,0.,0.,92.136,95.94,99.647,103.291,112.263,121.174,
     $130.113,0.,0.,91.137,95.1,98.921,102.652,111.771,120.775,129.778,
     $0.,0.,0.,0.,94.556,98.975,109.102,118.67,128.036/
C
C     SPECIFIC ENTROPY TABLE FOR SUPERHEATED R-12
C     TWO DIMENSION
C
      DATA SS/0.16996,0.17597,0.18172,0.18726,0.19262,0.19782,0.21022,
     $0.22191,1.,0.16559,0.17184,0.17778,0.18346,0.18892,0.19421,0.20677
```

```
     $,0.21856,1.,0.,0.16808,0.17423,0.18007,0.18566,0.19104,0.20377,
     $0.21567,1.,0.,0.16454,0.17094,0.17697,0.1827,0.18819,0.2011,
     $0.21312,1.,0.,0.,0.16783,0.17407,0.17995,0.18556,0.19868,0.21081,
     $0.2222,0.,0.,0.1648,0.1713,0.17737,0.18311,0.19644,0.2087,0.22017,
     $0.,0.,0.,0.,0.16537,0.17217,0.18697,0.2,0.21195/
C
C     TEMP. TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA T/10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.,75.,80.
     $,85.,90.,95.,100.,105.,110.,115.,120.,125.,130.,140.,150.,160./
C
C     PRESSURE TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA P/29.335,32.415,35.736,39.310,43.148,47.263,51.667,56.373,
     $61.394,66.743,72.433,78.477,84.888,91.682,98.87,106.47,114.49,
     $122.95,131.86,141.25,151.11,161.47,172.35,183.76,195.71,221.32,
     $249.31,279.82/
C
C     SPECIFIC GAS VOLUME TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA VG/1.3241,1.205,1.0988,1.0039,0.9188,0.84237,0.77357,0.71149,
     $0.65537,0.60453,0.55839,0.51642,0.47818,0.44327,0.41135,0.38212,
     $0.35529,0.33063,0.30794,0.28701,0.26769,0.24982,0.23326,0.21791,
     $0.20364,0.17799,0.15564,0.13604/
C
C     SPECIFIC LIQUID ENTHALPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA HF/10.684,11.771,12.863,13.958,15.058,16.163,17.273,18.387,
     $19.507,20.634,21.766,22.905,24.05,25.204,26.365,27.534,28.713,
     $29.901,31.1,32.31,33.531,34.765,36.013,37.275,38.553,41.162,
     $43.85,46.633/
C
C     SPECIFIC GAS ENTHALPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA HG/78.3355,78.861,79.385,79.904,80.419,80.93,81.436,81.937,
     $82.433,82.924,83.409,83.887,84.359,84.825,85.282,85.732,86.174,
     $86.606,87.029,87.442,87.844,88.233,88.61,88.973,89.321,89.967,
     $90.534,91.006/
C
C     SPECIFIC GAS ENTROPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA SG/0.16798,0.16758,0.16719,0.16683,0.16648,0.16616,0.16586,
     $0.16557,0.1653,0.16504,0.16479,0.16456,0.16434,0.16412,0.16392,
     $0.16372,0.16353,0.16334,0.16315,0.16297,0.16279,0.1626,0.16241,
```

```
      $0.16222,0.16202,0.16159,0.1611,0.16053/
C
C      SPECIFIC LIQUID ENTROPY TABLE FOR SATURATED R-12
C      ONE DIMENSION
C
       DATA SF/.023954,.026243,.028515,.030772,.033013,.03524,.037453,
      $.039652,.041839,.044015,.04618,.048336,.050482,.05262,.054751,
      $.056877,.058997,.061113,.063227,.065339,.067451,.069564,.07168,
      $.0738,.075927,.080205,.084531,.088927/
       I=1
C
C      COMPARE T1 WITH T(I):TEMP. IN SATURATED TEMP. TABLE
C      IF SMALLER, INTERPOLATE FOR P1, H1, S1, V1 AND H1F
C
20     IF (T1.LE.T(I)) GO TO 30
       I=I+1
       GO TO 20
30     H1=HG(I-1)+(HG(I)-HG(I-1))*(T1-T(I-1))/(T(I)-T(I-1))
       H1F=HF(I-1)+(HF(I)-HF(I-1))*(T1-T(I-1))/(T(I)-T(I-1))
       P1=P(I-1)+(P(I)-P(I-1))*(T1-T(I-1))/(T(I)-T(I-1))
       S1=SG(I-1)+(SG(I)-SG(I-1))*(T1-T(I-1))/(T(I)-T(I-1))
       S1F=SF(I-1)+(SF(I)-SF(I-1))*(T1-T(I-1))/(T(I)-T(I-1))
       V1=VG(I-1)+(VG(I)-VG(I-1))*(T1-T(I-1))/(T(I)-T(I-1))
       J=1
C
C      COMPARE T3 WITH T(J):TEMP. IN SATURATED TEMP. TABLE
C      IF SMALLER, INTERPOLATE FOR P2, H3, S3 AND SG3
C      H4, S4 AND S2 CAN BE CALCULATED
C
40     IF (T3.LE.T(J)) GO TO 50
       J=J+1
       GO TO 40
50     P2=P(J-1)+(P(J)-P(J-1))*(T3-T(J-1))/(T(J)-T(J-1))
       H3=HF(J-1)+(HF(J)-HF(J-1))*(T3-T(J-1))/(T(J)-T(J-1))
       S3=SF(J-1)+(SF(J)-SF(J-1))*(T3-T(J-1))/(T(J)-T(J-1))
       SG3=SG(J-1)+(SG(J)-SG(J-1))*(T3-T(J-1))/(T(J)-T(J-1))
       H4=H3
       FR=(H4-H1F)/(H1-H1F)
       S4=FR*S1+(1.-FR)*S1F
       S2=S1
       K=1
C
C      COMPARE P2 WITH PS(K):PRESSURE IN SUPERHEATED PRESSURE TABLE
C      IF SMALLER, INTERPOLATE FOR VP(L), HP(L), SP(L)
C
60     IF (P2.LE.PS(K)) GO TO 70
       K=K+1
       GO TO 60
70     DO 80 L=1,9
       VP(L)=VS(L,(K-1))+(VS(L,K)-VS(L,(K-1)))*(P2-PS(K-1))/(PS(K)-
```

```
     $PS(K-1))
      HP(L)=HS(L,(K-1))+(HS(L,K)-HS(L,(K-1)))*(P2-PS(K-1))/(PS(K)-
     $PS(K-1))
      SP(L)=SS(L,(K-1))+(SS(L,K)-SS(L,(K-1)))*(P2-PS(K-1))/(PS(K)-
     $PS(K-1))
80    CONTINUE
      M=1
C
C     COMPARE S2 WITH SP(M): ENTROPY TABLE FROM PREVIOUS CALCULATION
C     IF SMALLER, INTERPOLATE FOR T2, H2, V2
C     T4, P4, P3 CAN BE CALAULATED
C
90    IF (S2.LE.SP(M)) GO TO 100
      M=M+1
      GO TO 90
100   T2=TS(M-1)+(TS(M)-TS(M-1))*(S2-SP(M-1))/(SP(M)-SP(M-1))
      V2=VP(M-1)+(VP(M)-VP(M-1))*(S2-SP(M-1))/(SP(M)-SP(M-1))
      H2=HP(M-1)+(HP(M)-HP(M-1))*(S2-SP(M-1))/(SP(M)-SP(M-1))
      T4=T1
      P4=P1
      P3=P2
C
C     CALCULATE WORK IN, HEAT IN, HEAT OUT PER CYCLE AND C.O.P.
C
      WIN=H2-H1
      QIN=H1-H4
      QOUT=H2-H3
      COP=QIN/WIN
C
C     CALCULATE WORK IN KW, HEAT IN BTU/MIN
C
      CR=V1/V2
      VA=VMAX
      VB=VA/CR
      VC=VCL
      VD=VC*CR
      VIN=VA-VD
      MIN=(VIN*NCY)/(V1*1728.)
      WDIN=WIN*MIN*RPM*60./3412.
      QDIN=QIN*MIN*RPM
      QDOUT=QOUT*MIN*RPM
C
      RESULTS(1)=QDIN
      RESULTS(2)=QDOUT
C
      MDC=QDOUT/(0.24*(TCOUT-TCIN))
      MDE=QDIN/(0.24*(TEIN-TEOUT))
C
C    CALCULATE POWER FOR FAN
C
```

```
C     CHANGE UNITS FROM IN/SEC TO FT/HR
C
      U=300.*UH
C
C     AVERAGE TEMP. AT INLET OF COND.
C
      TF=(TCIN+T3)/2.
      IA=1
C
C     COMPARE TF WITH T(I), IF .LE.
C     INTERPOLATE FOR V, K, PR,
C
200   IF (TF.LE.TA(IA)) GO TO 220
      IA=IA+1
      GO TO 200
220    V=VT(IA-1)+(VT(IA)-VT(IA-1))*(TF-TA(IA-1))/(TA(IA)-TA(IA-1))
      KCO=KT(IA-1)+(KT(IA)-KT(IA-1))*(TF-TA(IA-1))/(TA(IA)-TA(IA-1))
C
      PR=PRT(IA-1)+(PRT(IA)-PRT(IA-1))*(TF-TA(IA-1))/(TA(IA)-T(IA-1))
C
C     REYNOLDS NUMBER
C
      RE=U*D/(V*12.)
C
C     NUSSELT NUMBER
C
      NU=.332*(RE**.5)*(PR**(1./3.))
C
C     HEAT-TRANSFER COEFFICIENT
C
      H=(NU*KCO/D)*12.
C
      PI=4.*ATAN(1.)
C
C     TOTAL HEAT TRANSFER AREA PER FIN
C
      A=(2.*PI*R*LPI+2.*D**2+4.*D*TH-PI*R**2)/144.
C
C     HEAT TRANSFERED PER FIN (BTU/HR)
C
      QMAX=H*A*(T3-TCIN)
C
C     NUMBER OF FINS NEEDED
C
      RN=(QDOUT/QMAX)*60.
      N=INT(RN)+1
C
      RESULTS(4)=FLOAT(N)
C
C     CHANGE UNIT FROM IN/SEC TO M/SEC
```

```
C
      US=UH*.0254
C
C     CHANGE UNIT FROM LB/MIN TO KG/SEC
C
      MDCS=MDC*.00756
C
C     WORK NEEDED IN KW
C     CONDENSER FAN POWER
      W=MDCS*US**2/2000.
      EU=300.*EUH
      ETF=(TEIN+T1)/2.
      JA=1
225   IF (ETF.LE.TA(JA)) GO TO 230
      JA=JA+1
      GO TO 225
230   EV=VT(JA-1)+(VT(JA)-VT(JA-1))*(ETF-TA(JA-1))/(TA(JA)-TA(JA-1))
      EK=KT(JA-1)+(KT(JA)-KT(JA-1))*(ETF-TA(JA-1))/(TA(JA)-TA(JA-1))
      EPR=PRT(JA-1)+(PRT(JA)-PRT(JA-1))*(ETF-TA(JA-1))/(TA(JA)-TA(JA-1))
      ERE=(EU*ED)/(EV*12.)
      ENU=.332*(ERE**.5)*(EPR**(1./3.))
      EH=(ENU*EK/ED)*12.
      EA=(2.*PI*ER*EL+2.*ED**2+4.*ED*ETH-PI*ER**2)/144.
      QEMAX=EH*EA*(TEIN-T1)
      ERN=(QDIN/QEMAX)*60.
      NE=INT(ERN)+1
C
      RESULTS(3)=FLOAT(NE)
C
      EUS=EUH*.0254
      MDES=MDE*.00756
      WE=MDES*EUS**2/2000.
C
C     CALCULATE ALUMINUM NEEDED FOR THE FINS
C
      WALCO=(D**2-PI*R**2)*TH*N*0.1
      WALEV=(ED**2-PI*ER**2)*ETH*NE*0.1
      WAL=WALCO+WALEV
C
C     CYLINDER HEIGHT: 1.5*DP INCH
C     CYLINDER WALL THICKNESS: 0.5 INCH
C     VOLUME FOR CYLINDER WALL (CAST IRON):
C
      PI=4.*ATAN(1.)
      AC=PI*((DP+1.)**2-DP**2)/4.
      VCW=NCY*1.5*DP*AC
C
C     CRANK CASE LENGTH: NCY*(DP+1)+NCY+1 INCH
C     CRANK CASE WIDTH: 2DP+1 INCH
C     CRANK CASE HEIGHT: DP+1 INCH
```

```
C     CRANK CASE WALL THICKNESS: 0.5 INCH
C     VOLUME FOR CRANK CASE WALL (CAST IRON):
C
      ACC1=(NCY*(DP+1.)+NCY+1.)*(2.*DP+1.)
      ACC2=(NCY*(DP+1.)+NCY+1.)*(DP+1.)
      ACC3=(DP+1.)*(2.*DP+1.)
      VCCW=2.*(ACC1+ACC2+ACC3)-NCY*(DP+1.)**2*PI/4.
C
C     PISTON HEIGHT: DP INCH
C     VOLUME FOR PISTONS (CAST IRON):
C
      VPI=NCY*PI*DP**3/4.
C
C     COVER PLATE LENGTH: NCY*(DP+1)+NCY+1 INCH
C     COVER PLATE WIDTH: 2DP INCH
C     COVER PLATE THICKNESS: 1 INCH
C     VOLUME FOR COVER PLATE (CAST IRON):
C
      VCP=(NCY*(2.*DP+1.)+NCY+1.)*DP
C
C     DIAMETER FOR CONNECTING ROD: 1 INCH
C     LENGTH FOR CONNECTING ROD: NCY*(DP+1)+NCY+5
C     VOLUME FOR CONNECTING RODS (STEEL):
C
      VCR=(NCY*(DP+1.)+NCY+5.)*PI/4.
C
C     DENSITY FOR CAST IRON: .265 LB/IN**3
C     DENSITY FOR STEEL: .283 LB/IN**3
C
      WCAST=(VCW+VCCW+VPI+VCP)*.265
      WSTEE=VCR*.283
C
C     TOTAL COST FOR THE MATERIAL FOR COMPRESSOR
C
      CCASU=.02
      CSTEU=.02
      CUAL=.02
      CCOMM=WCAST*CCASU+WSTEE*CSTEU
C
      COSTAL=WAL*CUAL
      RESULTS(10)=CCOMM+COSTAL+200.+(3.77*DP**2)+80.*NCY
C
C     TRANSFER UNIT FROM KW TO HP
C
      WDINH=WDIN*1.341
      WH=W*1.341
      WEH=WE*1.341

C
C     DETERMINE MOTOR FOR COMPRESSOR
```

```
C
      ITH=1
250   IF(WDINH.LE.THP(ITH)) GO TO 260
      ITH=ITH+1
      GO TO 250
260   THPCO=THP(ITH)
C
      RESULTS(7)=THPCO
C
C     DETERMINE MOTOR FOR FANS FOR EVA. AND CON.
C
      JTH=1
270   IF(WEH.LE.THP(JTH)) GO TO 280
C
      JTH=JTH+1
      GO TO 270
280   THPF=THP(JTH)
C
      RESULTS(8)=THPF
C
      JTH=1
271   IF(WH.LE.THP(JTH)) GO TO 281
C
      JTH=JTH+1
      GO TO 271
281   RESULTS(9)=THP(JTH)
C
C    RESULTS ARRAY
C
C          1     Evap Heat Rate, B/m
C          2     Cond Heat Rate, B/m
C          3         No of Evap Fins
C          4         No of Cond Fins
C          5         Length Evap, In
C          6         Length Cond, In
C          7     Compressor Motor Hp
C          8       Evap Fan Motor Hp
C          9       Cond Fan Motor Hp
C         10      Acquisition Cost $
C         11         Design Life, Hr
C         12                MTTR, Hr
C         13          UNIT MTBF, Hr
C         14        Overall MTBF, Hr
C
      RESULTS(6)=RESULTS(4)*LPI
      RESULTS(5)=RESULTS(3)*EL
      RESULTS(11)=((5.E7/RPM)/60.)*2.**(-.05*T2+15.)
C     ***DESIGN LIFE DEPENDENCY ON T2 IS TAKEN FROM AMBROSE'S
C       HEAT PUMPS AND ELECTRIC HEATING***
      RESULTS(12)=10.*NCY+(RESULTS(5)+RESULTS(6))/1.E4
```

```
      RESULTS(13)=1000./NCY-RPM**2/1.E5
C     ***DO NOT PRINT TO SCREEN IF IN SUBROUTINE SENSITIVE****
      IF (AATEST.EQ.4.) GO TO 93
      DO 10 N=1,12
      NN=N+1
10    CALL TCSREAL(NN,69,0,9,RESULTS(N))
93    CONTINUE
C
C   TEMPORARILY LOAD INDEP VARIBLES FOR OPTIM
C     DEMAND IN BTU/MIN
C   **NEXT CALC. WILL TRUNCATE TO INTEGER--MIN. QTY. TO BE INCR. BY 1.
      IDEMAND=4.E4/RESULTS(1)
      IDEMAND=IDEMAND+1
      XINT=.10
C     SHORTAGE PENALTY UNITS ARE DOLLARS PER BTU/MINUTE NOT
C      EXPECTED TO BE DELIVERED.
C     SHORTAGE COST WILL BE EXPECTED NUMBER OF UNITS DOWN MULT.
C      BY COOLING CAPACITY PER UNIT, MULT. BY SHORTAGE PENALTY.
      SHORT=30.8
C     NOTE THAT ELECTRIC POWER COST OF $.08/KW-HR IS EQUIVALENT
C      TO $12.33/(BTU/MIN)--$30.8 IS APPROX. 2.5 x 12.33
      RFAC=5000.
      RFAOP=10000.
C                                                                 ^
C     IDEMAND = DEMAND                                            ^
C     XINT = INTerst rate                                         ^
C     SHORT = SHORTage penalty.                                   ^
C     RFAC = annual Repair FACilty cost.                          ^
C     RFAOP = Repair Facilty Annual OPerating cost.               ^
C                                                                 ^
      ACQ=RESULTS(10)+291.26+20.35*(RESULTS(7)+RESULTS(8)+RESULTS(9))
      SALV=0.
      IDLIFE=IFIX(RESULTS(11)/24./365.0)
C     *VALUE NOT TO EXCEED 20 YRS DUE TO ARRAY SIZES LATER IN PROG.*
      IF (IDLIFE.GT.20) THEN
      IDLIFE=20
      ENDIF
C     **LARGE EQUIPMENT LIFE MIGHT BE REASON TO HAVE SALVAGE VALUE*
      AOPFU=(WDINH+WH+WEH)*24.*365.25*.08
      AOPLA=RESULTS(12)*262800./(RESULTS(12)+RESULTS(13))
      AOPPM=(RESULTS(5)+RESULTS(6))/12.*.10
      AOPOTH=0.
      DO 300 INDEX=1,20
      XMTBF(INDEX)=RESULTS(13)/24./365.
300   XMTTR(INDEX)=RESULTS(12)/24./365.
C                                                                 ^
C     ACQ = ACQuisition cost.                                     ^
C     SALV = SALVage value at end of design life.                 ^
C     IDLIFE = Design LIFE of the equipment                       ^
C     AOPFU = Annual OPerating cost (FUel).                       ^
```

```
C     AOPLA = Annual OPerating cost (LAbor).                                ^
C     AOPPM = Annual OPerating cost (Preventive Maintenance).              ^
C     AOPOTH = Annual OPerating cost (OTHer).                               ^
C     XMTBF = array containing Mean Time Between Failures.                 ^
C     XMTTR = array containing Mean Time To Repair the equipment.          ^
C                                                                           ^
      EAETSC=0.
      CALL XOPT(EAETSC)
C
C     JN IS NUMBER DEPLOYED, FROM REPS ROUTINE, VIA XOPT
C     ES IS EXPECTED NUMBER OF UNITS SHORT
C     EXCOOL IS EXPECTED BTUs PER MINUTE REMOVED
C
      EXCOOL=(JN-ES)*RESULTS(1)
      RESULTS(14)=RESULTS(13)/(JN-ES)
C     ***OVERALL MTBF SHOULD HAVE EFFECT OF "BATHTUB" CURVE**
C     **DO NOT PRINT TO SCREEN IF IN SENSITIVE SUBROUTINE**
      IF (AATEST.EQ.4.) GO TO 99
      CALL TCSREAL(14,69,0,9,RESULTS(14))
      CALL TCSREAL(17,69,0,9,EXCOOL)
      CALL TCSREAL(20,69,0,9,EAETSC)
C     ***DISPLAY RESULT VS. PREVIOUS TRY***
C        BUT DO NOT CHANGE SIGNS IF NO DATA CHANGES (e.g. IF REVIEW
C         OR SENSITIVE HAVE BEEN CALLED.)
      IF (BBTEST.EQ.0.) GO TO 121
      IF (YY.EQ.1.) THEN
      CALL TCSTXT(19,77,0,2,1,'+')
      END IF
      IF (YY.EQ.0.) THEN
      CALL TCSTXT(19,77,0,2,1,'0')
      END IF
      IF (YY.EQ.-1.) THEN
      CALL TCSTXT(19,77,0,2,1,'-')
      END IF
C
      IF (ZZ.EQ.1.) THEN
      CALL TCSTXT(17,77,0,2,1,'+')
      END IF
      IF (ZZ.EQ.0.) THEN
      CALL TCSTXT(17,77,0,2,1,'0')
      END IF
      IF (ZZ.EQ.-1.) THEN
      CALL TCSTXT(17,77,0,2,1,'-')
      END IF
121   CONTINUE
C
      IF (BBTEST.EQ.2..OR.BBTEST.EQ.4.) GO TO 111
      IF (EAETSC.GT.REFCST) THEN
      YY=1.
      CALL TCSTXT(19,77,0,2,1,'+')
```

```
      END IF
      IF (EAETSC.EQ.REFCST) THEN
      YY=0.
      CALL TCSTXT(19,77,0,2,1,'0')
      END IF
      IF (EAETSC.LT.REFCST) THEN
      YY=-1.
      CALL TCSTXT(19,77,0,2,1,'-')
      END IF
C
      IF (EXCOOL.GT.RFCOOL) THEN
      ZZ=1.
      CALL TCSTXT(17,77,0,2,1,'+')
      END IF
      IF (EXCOOL.EQ.RFCOOL) THEN
      ZZ=0.
      CALL TCSTXT(17,77,0,2,1,'0')
      END IF
      IF (EXCOOL.LT.RFCOOL) THEN
      ZZ=-1.
      CALL TCSTXT(17,77,0,2,1,'-')
      END IF
C
      REFCST=EAETSC
C     **REFCST IS REFERENCE COST** SAVES EAETSC FOR COMPARISON TO
C        PREVIOUS CALCULATION.  LOCATED HERE IN PROGRAM IN ORDER TO
C        BE SUPPRESSED IF IN SUBROUTINE SENSITIVE.
      RFCOOL=EXCOOL
C     **RFCOOL IS REFERENCE EXPECTED COOLING VALUE FOR COMPARISON**
C
111   CONTINUE
      CALL TCSREAL(15,69,0,9,ES)
      CALL TCSTXT(15,47,0,2,19,'  Expected Shortage')
99    CONTINUE
      BBTEST=1.
C     ***SETS CONDITION TO HOLD +,0, OR - SIGNS FOR EXCOOL AND
C         EAETSC IF LEAVING EDIT FOR SUBROUTINE REVIEW OR SUBROUTINE
C         SENSITIVE.  (SO THAT WHEN EDIT SCREEN COMES BACK, THE
C         SIGNS WILL NOT DEFAULT TO ZERO.
      RETURN
      END
C
C
      SUBROUTINE PLOT(ITYPE)
C
C     ITYPE = 1 T-S
C             2 P-H
C             3 P-V
C
C     SUBROUTINE FOR PLOTTING T-S, P-H AND P-V DIAGRAMS
```

```
C
C   1      T1      EVAP TEMP, DEG F
C   2      T3      COND TEMP, DEG F
C   3      NCY     NO OF CYLS
C   4      VCL     CL VOL/CYL, CUBIC IN
C   5      VMAX    MAX VOL/CYL, CUBIC IN
C   6      RPM     CRANK SPEED, RPM
C   7      TCIN    COND INLET AIR TEMP, DEG F
C   8      TCOUT   COND EXH AIR TEMP, DEG F
C   9      TEIN    EVAP INLET AIR TEMP, DEG F
C   10     TEOUT   EVAP EXH AIR TEMP, DEG F
C   11     D       COND FIN WIDTH, IN
C   12     TH      COND FIN THICKNESS, IN
C   13     R       COND TUBE RADIUS, IN
C   14     LPI     COND FIN PITCH, IN
C   15     UH      COND AIR VELOCITY, IN/S
C   16     ED      EVAP FIN WIDTH,IN
C   17     ETH     EVAP FIN THICKNESS, IN
C   18     ER      EVAP TUBE RADIUS, IN
C   19     EL      EVAP FIN PITCH, IN
C   20     EUH     EVAP AIR VELOCITY, IN/S
C   21     DP      PISTON DIA, IN
C
C      COMMON/CINP/T1,T3,NCY,VCL,VMAX,RPM,TCIN,TCOUT,TEIN,TEOUT,
C     $D,TH,R,LPI,UH,ED,ETH,ER,EL,EUH,DP,CCASU,CSTEU,CUAL
C
       COMMON/CINP/NCY,RPM,DP,VMAX,VCL,T3,TCIN,TCOUT,TEIN,TEOUT,
      $ D,TH,LPI,R,UH,ED,ETH,EL,ER,EUH,T1
C
       COMMON/COUT/S1,P1,V1,H1,T2,S2,P2,V2,H2,S3,H3,H4,S4,WIN,QIN,
      $QOUT,COP,CR,MIN,WDIN,QDIN,QDOUT,MDC,MDE,N,W,NE,WE,WCAST,WSTEE,
      $CCOMM,THPCO,THPF,SG3, VA, VB, VC, VD,T4,P3,P4,CCM,CFM,COSTAL,WAL
C
       DIMENSION T(28), P(28), VG(28), HF(28), HG(28), SG(28), SF(28)
      $,PS(7), TS(9), VS(9,7), HS(9,7), SS(9,7), VP(9), HP(9), SP(9)
      $,YT(7), XS(7), XH(6), YP(6), XV(16), YCP(16), XSF(29), XSG(29)
      $,YST(29),LAB1(11),LAB2(7),LAB5(6),LAB6(8),XHF(29),XHG(29),
      $YSP(29),LAB3(8),IT1(9),IT2(9),IT3(9),IT4(9),IS1(9),IS2(9),
      $IS3(9),IS4(9),IP1(9),IP2(9),IP3(9),IP4(9),IH1(9),IH2(9),
      $IH3(9),IH4(9),ISG3(9)
       DATA LAB1/84,69,77,80,69,82,65,84,85,82,69/
       DATA LAB2/69,78,84,82,79,80,89/
       DATA LAB3/69,78,84,72,65,76,80,89/
       DATA LAB5/86,79,76,86,77,69/
       DATA LAB6/80,82,69,83,83,85,82,69/
C
C      PRESSURE TABLE FOR SUPERHEATED R-12
C      ONE DIMENSION
C
       DATA PS/100.,120.,140.,160.,180.,200.,300./
```

```
C
C     TEMP. TABLE FOR SUPERHEATED R-12
C     ONE DIMENSION
C
      DATA TS/100.,120.,140.,160.,180.,200.,250.,300.,350./
C
C     SPECIFIC VOLUME TABLE FOR SUPERHEATED R-12
C     TWO DIMENSION
C
      DATA VS/0.43138,0.45562,0.47881,0.50118,0.52291,0.54413,0.59549,
     $0.64518,1.,0.34655,0.36841,0.38901,0.4087,0.42766,0.44606,0.49025,
     $0.53267,1.,0.,0.30549,0.32445,0.34232,0.35939,0.37584,0.41499,
     $0.45226,1.,0.,0.25764,0.27558,0.29224,0.30797,0.32301,0.35847,
     $0.3919,1.,0.,0.,0.2371,0.25297,0.26775,0.28176,0.31142,0.34492,
     $0.37409,0.,0.,0.20579,0.22121,0.23535,0.2486,0.27911,0.3073,
     $0.33408,0.,0.,0.,0.,0.13482,0.14697,0.1723,0.19402,0.21388/
C
C     SPECIFIC ENTHALPY TABLE FOR SUPERHEATED R-12
C     TWO DIMENSION
C
      DATA HS/88.694,92.116,95.507,98.884,102.257,105.633,114.119,
     $122.707,1000.,87.675,91.237,94.736,98.199,101.642,105.076,113.67,
     $122.333,1000.,0.,90.297,93.923,97.483,101.003,104.501,113.212,
     $121.953,1000.,0.,89.283,93.059,96.732,100.34,103.907,112.743,
     $121.567,1000.,0.,0.,92.136,95.94,99.647,103.291,112.263,121.174,
     $130.113,0.,0.,91.137,95.1,98.921,102.652,111.771,120.775,129.778,
     $0.,0.,0.,0.,94.556,98.975,109.102,118.67,128.036/
C
C     SPECIFIC ENTROPY TABLE FOR SUPERHEATED R-12
C     TWO DIMENSION
C
      DATA SS/0.16996,0.17597,0.18172,0.18726,0.19262,0.19782,0.21022,
     $0.22191,1.,0.16559,0.17184,0.17778,0.18346,0.18892,0.19421,0.20677
     $,0.21856,1.,0.,0.16808,0.17423,0.18007,0.18566,0.19104,0.20377,
     $0.21567,1.,0.,0.16454,0.17094,0.17697,0.1827,0.18819,0.2011,
     $0.21312,1.,0.,0.,0.16783,0.17407,0.17995,0.18556,0.19868,0.21081,
     $0.2222,0.,0.,0.1648,0.1713,0.17737,0.18311,0.19644,0.2087,0.22017,
     $0.,0.,0.,0.,0.16537,0.17217,0.18697,0.2,0.21195/
C
C     TEMP. TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA T/10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.,75.,80.
     $,85.,90.,95.,100.,105.,110.,115.,120.,125.,130.,140.,150.,160./
C
C     PRESSURE TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA P/29.335,32.415,35.736,39.310,43.148,47.263,51.667,56.373,
     $61.394,66.743,72.433,78.477,84.888,91.682,98.87,106.47,114.49,
```

```
     $122.95,131.86,141.25,151.11,161.47,172.35,183.76,195.71,221.32,
     $249.31,279.82/
C
C     SPECIFIC GAS VOLUME TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA VG/1.3241,1.205,1.0988,1.0039,0.9188,0.84237,0.77357,0.71149,
     $0.65537,0.60453,0.55839,0.51642,0.47818,0.44327,0.41135,0.38212,
     $0.35529,0.33063,0.30794,0.28701,0.26769,0.24982,0.23326,0.21791,
     $0.20364,0.17799,0.15564,0.13604/
C
C     SPECIFIC LIQUID ENTHALPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA HF/10.684,11.771,12.863,13.958,15.058,16.163,17.273,18.387,
     $19.507,20.634,21.766,22.905,24.05,25.204,26.365,27.534,28.713,
     $29.901,31.1,32.31,33.531,34.765,36.013,37.275,38.553,41.162,
     $43.85,46.633/
C
C     SPECIFIC GAS ENTHALPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA HG/78.3355,78.861,79.385,79.904,80.419,80.93,81.436,81.937,
     $82.433,82.924,83.409,83.887,84.359,84.825,85.282,85.732,86.174,
     $86.606,87.029,87.442,87.844,88.233,88.61,88.973,89.321,89.967,
     $90.534,91.006/
C
C     SPECIFIC GAS ENTROPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA SG/0.16798,0.16758,0.16719,0.16683,0.16648,0.16616,0.16586,
     $0.16557,0.1653,0.16504,0.16479,0.16456,0.16434,0.16412,0.16392,
     $0.16372,0.16353,0.16334,0.16315,0.16297,0.16279,0.1626,0.16241,
     $0.16222,0.16202,0.16159,0.1611,0.16053/
C
C     SPECIFIC LIQUID ENTROPY TABLE FOR SATURATED R-12
C     ONE DIMENSION
C
      DATA SF/.023954,.026243,.028515,.030772,.033013,.03524,.037453,
     $.039652,.041839,.044015,.04618,.048336,.050482,.05262,.054751,
     $.056877,.058997,.061113,.063227,.065339,.067451,.069564,.07168,
     $.0738,.075927,.080205,.084531,.088927/
C
C     BRANCH TO T-S, P-H, OR P-V SECTIONS
C
      GO TO (600,400,500), ITYPE
C
C    T-S DIAGRAM
C
600   XS(1)=6.
```

```
      XS(2)=S1
      XS(3)=S2
      XS(4)=SG3
      XS(5)=S3
      XS(6)=S4
      XS(7)=S1
      YT(1)=6.
      YT(2)=T1
      YT(3)=T2
      YT(4)=T3
      YT(5)=T3
      YT(6)=T1
      YT(7)=T1
      XSG(1)=28.
      XSF(1)=28.
      YST(1)=28.
      DO 220 IT=1, 28
      XSG(IT+1)=SG(IT)
      XSF(IT+1)=SF(IT)
      YST(IT+1)=T(IT)
220   CONTINUE
      L1=11
      L2=7
      AMIN=SF(1)
      AMAX=.17
      BMIN=T(1)
      IF (T2.GT.T(28)) MAX=T2
      BMAX=T(28)
      CALL INITT (120)
      CALL BINITT
      CALL XNEAT(0)
      CALL YNEAT(0)
      CALL MNMX (XS,AMIN,AMAX)
      CALL MNMX (XSG,AMIN,AMAX)
      CALL MNMX (XSF,AMIN,AMAX)
      CALL MNMX (YT,BMIN,BMAX)
      CALL MNMX (YST,BMIN,BMAX)
      CALL DLIMX (AMIN,AMAX)
      CALL DLIMY (BMIN,BMAX)
      CALL SLIMX(150,700)
      CALL CHECK (XSF,YST)
      CALL DSPLAY (XSF,YST)
      CALL CPLOT (XSG,YST)
      CALL SYMBL(1)
      CALL CPLOT (XS,YT)
      DO 700 ITS=2,6
      CALL MOVEA(XS(ITS),YT(ITS))
      CALL MOVREL(-40,20)
      JTS=ITS+47
      CALL SYMOUT(JTS,1.)
```

```
700   CONTINUE
      CALL MOVABS(1,600)
      CALL VLABEL(L1,LAB1)
      CALL MOVABS(200,1)
      CALL HLABEL(L2,LAB2)
      CALL MOVABS(800,720)
      CALL SYMOUT(84,2.)
      CALL MOVABS(920,720)
      CALL SYMOUT(83,2.)
      CALL MOVABS(750,600)
      CALL SYMOUT(49,2.)
      CALL MOVABS(750,500)
      CALL SYMOUT(50,2.)
      CALL MOVABS(750,400)
      CALL SYMOUT(51,2.)
      CALL MOVABS(750,300)
      CALL SYMOUT(52,2.)
      CALL MOVABS(750,200)
      CALL SYMOUT(53,2.)
      CALL FFORM(T1,9,5,IT1,32)
      CALL NOTATE(770,600,9,IT1)
      CALL FFORM(T2,9,5,IT2,32)
      CALL NOTATE(770,500,9,IT2)
      CALL FFORM(T3,9,5,IT3,32)
      CALL NOTATE(770,400,9,IT3)
      CALL FFORM(T3,9,5,IT3,32)
      CALL NOTATE(770,300,9,IT3)
      CALL FFORM(T4,9,5,IT4,32)
      CALL NOTATE(770,200,9,IT4)
      CALL FFORM(S1,9,5,IS1,32)
      CALL NOTATE(900,600,9,IS1)
      CALL FFORM(S2,9,5,IS2,32)
      CALL NOTATE(900,500,9,IS2)
      CALL FFORM(SG3,9,5,ISG3,32)
      CALL NOTATE(900,400,9,ISG3)
      CALL FFORM(S3,9,5,IS3,32)
      CALL NOTATE(900,300,9,IS3)
      CALL FFORM(S4,9,5,IS4,32)
      CALL NOTATE(900,200,9,IS4)
      CALL TCSKEY (ISCAN,IASC,NCHAR)
      RETURN
C
C     DRAW P-H DIAGRAM ON SCREEN
C
400   CONTINUE
      XH(1)=5.
      XH(2)=H1
      XH(3)=H2
      XH(4)=H3
      XH(5)=H4
```

```
      XH(6)=H1
      YP(1)=5.
      YP(2)=P1
      YP(3)=P2
      YP(4)=P2
      YP(5)=P1
      YP(6)=P1
      XHF(1)=28.
      XHG(1)=28.
      YSP(1)=28.
      DO 230 IP=1,28
      XHF(IP+1)=HF(IP)
      XHG(IP+1)=HG(IP)
      YSP(IP+1)=P(IP)
230   CONTINUE
      CMIN=P(1)
      CMAX=P(28)
      DMIN=HF(1)
      DMAX=H2+10.
      CALL INITT (120)
      CALL BINITT
      CALL XNEAT(0)
      CALL YNEAT(0)
      CALL MNMX(XH,DMIN,DMAX)
      CALL MNMX(XHF,DMIN,DMAX)
      CALL MNMX(XHG,DMIN,DMAX)
      CALL MNMX(YP,CMIN,CMAX)
      CALL MNMX(YSP,CMIN,CMAX)
      CALL DLIMX(DMIN,DMAX)
      CALL DLIMY(CMIN,CMAX)
      CALL SLIMX(150,700)
      CALL CHECK (XHF,YSP)
      CALL DSPLAY (XHF,YSP)
      CALL CPLOT(XHG,YSP)
      CALL SYMBL(1)
      CALL CPLOT(XH,YP)
      DO 800 IPH=2,5
      CALL MOVEA(XH(IPH),YP(IPH))
      CALL MOVREL(-40,20)
      JPH=IPH+47
      CALL SYMOUT(JPH,1.)
800   CONTINUE
      L3=8
      L4=8
      CALL MOVABS(1,600)
      CALL VLABEL(L3,LAB6)
      CALL MOVABS(200,1)
      CALL HLABEL(L4,LAB3)
      CALL MOVABS(800,720)
      CALL SYMOUT(80,2.)
```

```
      CALL MOVABS(920,720)
      CALL SYMOUT(72,2.)
      CALL MOVABS(750,600)
      CALL SYMOUT(49,2.)
      CALL MOVABS(750,500)
      CALL SYMOUT(50,2.)
      CALL MOVABS(750,400)
      CALL SYMOUT(51,2.)
      CALL MOVABS(750,300)
      CALL SYMOUT(52,2.)
      CALL FFORM(P1,9,5,IP1,32)
      CALL NOTATE(770,600,9,IP1)
      CALL FFORM(P2,9,5,IP2,32)
      CALL NOTATE(770,500,9,IP2)
      CALL FFORM(P3,9,5,IP3,32)
      CALL NOTATE(770,400,9,IP3)
      CALL FFORM(P4,9,5,IP4,32)
      CALL NOTATE(770,300,9,IP4)
      CALL FFORM(H1,9,5,IH1,32)
      CALL NOTATE(890,600,9,IH1)
      CALL FFORM(H2,9,5,IH2,32)
      CALL NOTATE(890,500,9,IH2)
      CALL FFORM(H3,9,5,IH3,32)
      CALL NOTATE(890,400,9,IH3)
      CALL FFORM(H4,9,5,IH4,32)
      CALL NOTATE(890,300,9,IH4)
      CALL TCSKEY (ISCAN,IASC,NCHAR)
      RETURN
C
C     DRAW P-V DIAGRAM ON SCREEN
C
500   CONTINUE
      CON=(LOG(P1/P2))/(LOG(VB/VA))
      XV(1)=15.
      XV(2)=VA
      YCP(1)=15.
      YCP(2)=P1
      DO 300 IV=1,5
      XV(IV+2)=VA-(VA-VB)*IV/6.
      YCP(IV+2)=P1*(VA/XV(IV+2))**CON
300   CONTINUE
      XV(8)=VB
      XV(9)=VC
      YCP(8)=P2
      YCP(9)=P2
      DO 350 JV=1,5
      XV(JV+9)=VC+(VD-VC)*JV/6.
      YCP(JV+9)=P2*(VC/XV(JV+9))**CON
350   CONTINUE
      XV(15)=VD
```

```
      XV(16)=VA
      YCP(15)=P1
      YCP(16)=P1
      CALL INITT (120)
      CALL BINITT
      CALL CHECK (XV,YCP)
      CALL DSPLAY (XV,YCP)
      L5=6
      L6=8
      CALL MOVABS(200,1)
      CALL HLABEL(L5,LAB5)
      CALL MOVABS(1,600)
      CALL VLABEL(L6,LAB6)
      CALL TCSKEY (ISCAN,IASC,NCHAR)
      RETURN
      END
C
C
      SUBROUTINE REVIEW
C                                                                      ^
      COMMON/INDEP/ IDEMAND, XINT, SHORT, RFAC, RFAOP
      COMMON BBTEST
C                                                                      ^
C     IDEMAND = DEMAND                                                 ^
C     XINT = INTerst rate                                              ^
C     SHORT = SHORTage penalty.                                        ^
C     RFAC = annual Repair FACilty cost.                               ^
C     RFAOP = Repair Facility Annual OPerating cost.                   ^
C                                                                      ^
      COMMON/DEP/ACQ,SALV,IDLIFE,AOPFU,AOPLA,AOPPM,AOPOTH,XMTBF(20)
     1 ,XMTTR(20),ES
C                                                                      ^
C     ACQ = ACQuisition cost.                                          ^
C     SALV = SALVage value at end of design life.                      ^
C     IDLIFE = Design LIFE of the equipment                            ^
C     AOPFU = Annual OPerating cost (FUel).                            ^
C     AOPLA = Annual OPerating cost (LAbor).                           ^
C     AOPPM = Annual OPerating cost (Preventive Maintenance).          ^
C     AOPOTH = Annual OPerating cost (OTHer).                          ^
C     XMTBF = array containing Mean Time Between Failures.             ^
C     XMTTR = array containing Mean Time To Repair the equipment.      ^
C                                                                      ^
      CHARACTER*9 DAY9,DATE9
      CHARACTER*8 TIME8
      CHARACTER*40 LABEL40(15)
      COMMON/REPS/IM,INN,JN
C
      DATA LABEL40/
     1 '                 Cooling Demand, Btu/min',
     2 '                      Interest Rate, %',
```

```
     3 '             Shortage Penalty, $/(Btu/min)',
     4 '            Annual Repair Facility Cost, $',
     5 'Repair Facility Annual Operating Cost, $',
     6 '                  Unit Acquisition Cost, $',
     7 '                          Salvage Value, $',
     8 '                          Design Life, Yrs',
     9 '                       Annual Fuel Cost, $',
     * '                             Labor Cost, $',
     1 '                  Other Operating Costs, $',
     2 '   Number of Units Needed to Meet Demand',
     3 '                  Number of Units Deployed',
     4 '                 Number of Repair Channels',
     5 '                        Economic Life, Yrs'/
C
      CALL TCSMOD(0)
      CALL TCSSCR(0,0,24,79,0,0)
C
C   PUT UP COLUMN HEADERS
C
C      CALL DAY(DAY9)
C      CALL DATE(DATE9)
C      CALL TIME(TIME8)
      CALL TCSTXT(1,32,0,14,9,DAY9)
      CALL TCSTXT(1,42,0,14,9,DATE9)
      CALL TCSTXT(1,52,0,14,8,TIME8)
      CALL TCSTXT(1,10,0,14,25,'Parameter/Variable Review')
C
      DO 101 N=8,61
      CALL TCSTXT( 0,N,0, 9,1,'M')
      CALL TCSTXT( 2,N,0, 9,1,'M')
      CALL TCSTXT( 4,N,0, 9,1,'M')
      CALL TCSTXT( 9,N,0, 9,1,'M')
      CALL TCSTXT(17,N,0, 9,1,'M')
      CALL TCSTXT(21,N,0, 9,1,'M')
101   CONTINUE
C
      DO 102 N=0,21
      CALL TCSTXT(N, 8,0,9,1,':')
      CALL TCSTXT(N,61,0,9,1,':')
      IF(N.GE.3) CALL TCSTXT(N,49,0,9,1,'3')
102   CONTINUE
C
      CALL TCSTXT( 0,8,0, 9,1,'I')
      CALL TCSTXT( 2,8,0, 9,1,'L')
      CALL TCSTXT( 4,8,0, 9,1,'L')
      CALL TCSTXT( 9,8,0, 9,1,'L')
      CALL TCSTXT(17,8,0, 9,1,'L')
      CALL TCSTXT(21,8,0, 9,1,'H')
C
      CALL TCSTXT( 0,61,0, 9,1,';')
```

```
      CALL TCSTXT( 2,61,0, 9,1,'9')
      CALL TCSTXT( 4,61,0, 9,1,'9')
      CALL TCSTXT( 9,61,0, 9,1,'9')
      CALL TCSTXT(17,61,0, 9,1,'9')
      CALL TCSTXT(21,61,0, 9,1,'<')
C
      CALL TCSTXT( 2,49,0,9,1,'Q')
      CALL TCSTXT( 4,49,0,9,1,'X')
      CALL TCSTXT( 9,49,0,9,1,'X')
      CALL TCSTXT(17,49,0,9,1,'X')
      CALL TCSTXT(21,49,0,9,1,'O')
C
      CALL TCSTXT( 5,63,0,2, 6,'DESIGN')
      CALL TCSTXT( 6,63,0,2,11,'INDEPENDENT')
      CALL TCSTXT( 7,63,0,2,10,'PARAMETERS')
C
      CALL TCSTXT(12,63,0,2, 6,'DESIGN')
      CALL TCSTXT(13,63,0,2, 9,'DEPENDENT')
      CALL TCSTXT(14,63,0,2,10,'PARAMETERS')
C
      CALL TCSTXT(18,63,0,2, 6,'DESIGN')
      CALL TCSTXT(19,63,0,2, 9,'VARIABLES')
C
      CALL TCSTXT( 3,9,0,2,40,LABEL40( 1))
      CALL TCSTXT( 5,9,0,2,40,LABEL40( 2))
      CALL TCSTXT( 6,9,0,2,40,LABEL40( 3))
      CALL TCSTXT( 7,9,0,2,40,LABEL40( 4))
      CALL TCSTXT( 8,9,0,2,40,LABEL40( 5))
      CALL TCSTXT(10,9,0,2,40,LABEL40( 6))
      CALL TCSTXT(11,9,0,2,40,LABEL40( 7))
      CALL TCSTXT(12,9,0,2,40,LABEL40( 8))
      CALL TCSTXT(13,9,0,2,40,LABEL40( 9))
      CALL TCSTXT(14,9,0,2,40,LABEL40(10))
      CALL TCSTXT(15,9,0,2,40,LABEL40(11))
      CALL TCSTXT(16,9,0,2,40,LABEL40(12))
      CALL TCSTXT(18,9,0,2,40,LABEL40(13))
      CALL TCSTXT(19,9,0,2,40,LABEL40(14))
      CALL TCSTXT(20,9,0,2,40,LABEL40(15))
C
      DEMAND=IDEMAND
      DLIFE=IDLIFE
      XM=IM
      XN=JN
      XNN=INN
      YINT=XINT*100.
      CALL TCSTXT( 3,50,0,14,5,'40000')
      CALL TCSREAL( 5,50,0,14,YINT)
      CALL TCSREAL( 6,50,0,14,SHORT)
      CALL TCSREAL( 7,50,0,14,RFAC)
      CALL TCSREAL( 8,50,0,14,RFAOP)
```

```
      CALL TCSREAL(10,50,0,14,ACQ)
      CALL TCSREAL(11,50,0,14,SALV)
      CALL TCSREAL(12,50,0,14,DLIFE)
      CALL TCSREAL(13,50,0,14,AOPFU)
      CALL TCSREAL(14,50,0,14,AOPLA)
      CALL TCSREAL(15,50,0,14,AOPOTH)
      CALL TCSREAL(16,50,0,14,DEMAND)
      CALL TCSREAL(18,50,0,14,XN)
      CALL TCSREAL(19,50,0,14,XM)
      CALL TCSREAL(20,50,0,14,XNN)
C
      CALL TCSKEY(ISCAN,IASC,NCHAR)
      BBTEST=2.
C     ***SETS UP TEST TO KEEP +,0,- COMPARISON SIGNS FROM
C         SUBROUTINE EDIT FROM DEFAULTING TO ZERO WHEN NOT EDITING.
      RETURN
      END
C
C
      SUBROUTINE SENSITIVE
      REAL*4 DATA
      COMMON /CINP/ DATA(21)
      COMMON TEMP(21)
      COMMON EAETSC
      COMMON AATEST
      COMMON BBTEST
      CHARACTER*33 LABEL33(21)
      DIMENSION IRANK(21)
      REAL B(21)
C
      DATA LABEL33/'                  Number of Cylinders',
     1 '                 Crank Speed, RPM','                 Piston Diamete
     2r, In','   Maximum Cylinder Volume, Cu In',
     3 ' Cylinder Clearance Volume, Cu In',
     4 'Condenser-Mean Temperature, Deg F',
     5 ' Condenser, Inlet Air Temp, Deg F',
     6 '  Condenser, Exit Air Temp, Deg F',
     7 'Evaporator, Inlet Air Temp, Deg F',
     8 ' Evaporator, Exit Air Temp, Deg F',
     9 ' Condenser      Fin Width, In (C)',
     * ' Condenser  Fin Thickness, In (C)',
     1 ' Condenser      Fin Pitch, In (C)',
     2 ' Condenser    Tube Radius, In (C)',
     3 ' Condenser  Air Velocity, fpm (C)',
     4 ' Evaporator     Fin Width, In (E)',
     5 ' Evaporator Fin Thickness, In (E)',
     6 ' Evaporator     Fin Pitch, In (E)',
     7 ' Evaporator   Tube Radius, In (E)',
     8 ' Evaporator Air Velocity, fpm (E)',
     9 'Evaporator-Mean Temperature,Deg F'/
```

```
C
C   1    NCY    NO OF CYLS
C   2    RPM    CRANK SPEED, RPM
C   3    DP     PISTON DIA, IN
C   4    VMAX   MAX VOL/CYL, CUBIC IN
C   5    VCL    CL VOL/CYL, CUBIC IN
C   6    T3     COND TEMP, DEG F
C   7    TCIN   COND INLET AIR TEMP, DEG F
C   8    TCOUT  COND EXH AIR TEMP, DEG F
C   9    TEIN   EVAP INLET AIR TEMP, DEG F
C  10    TEOUT  EVAP EXH AIR TEMP, DEG F
C  11    D      COND FIN WIDTH, IN
C  12    TH     COND FIN THICKNESS, IN
C  13    LPI    COND FIN PITCH, IN
C  14    R      COND TUBE RADIUS, IN
C  15    UH     COND AIR VELOCITY, IN/S
C  16    ED     EVAP FIN WIDTH,IN
C  17    ETH    EVAP FIN THICKNESS, IN
C  18    EL     EVAP FIN PITCH, IN
C  19    ER     EVAP TUBE RADIUS, IN
C  20    EUH    EVAP AIR VELOCITY, IN/S
C  21    T1     EVAP TEMP, DEG F
C
      CALL TCSMOD(0)
      CALL TCSSCR(0,0,24,79,0,0)
C
C    PUT UP COLUMN HEADERS
C
      CALL TCSTXT(1,25,0,14,26,'Design Feature Sensitivity')
C
      DO 101 N=15,68
      CALL TCSTXT( 0,N,0, 9,1,'M')
      CALL TCSTXT( 2,N,0, 9,1,'M')
      CALL TCSTXT(24,N,0, 9,1,'M')
101   CONTINUE
C
      DO 102 N=0,24
      CALL TCSTXT( N,15,0,9,1,':')
      CALL TCSTXT( N,68,0,9,1,':')
      IF(N.GE.3.AND.N.LE.23) THEN
                             CALL TCSTXT( N,49,0,9,1,'3')
                             CALL TCSTXT( N,52,0,9,1,'3')
                             CALL TCSTXT( N,64,0,9,1,'3')
                             ENDIF
      IF(N.GE.3.AND.N.LE.23) THEN
                             NN=N-2
                             CALL TCSTXT(N,16,0,3,33,LABEL33(NN))
                             CALL TCSREAL(N,53,0,14,DATA(NN))
                             ENDIF
102   CONTINUE
```

```
C
      CALL TCSTXT( 0,15,0, 9,1,'I')
      CALL TCSTXT( 2,15,0, 9,1,'L')
      CALL TCSTXT(24,15,0, 9,1,'H')
C
      CALL TCSTXT( 0,68,0, 9,1,';')
      CALL TCSTXT( 2,68,0, 9,1,'9')
      CALL TCSTXT(24,68,0, 9,1,'<')
C
      CALL TCSTXT( 2,49,0,9,1,'Q')
      CALL TCSTXT(24,49,0,9,1,'O')
      CALL TCSTXT( 2,52,0,9,1,'Q')
      CALL TCSTXT(24,52,0,9,1,'O')
      CALL TCSTXT( 2,64,0,9,1,'Q')
      CALL TCSTXT(24,64,0,9,1,'O')
C
C    COMPUTE SENSITIVITY RANKINGS
C     *****SENSITIVITY SHOULD BE BASED ON DIRECTIONAL DERIVATIVES*
C      SENSITIVITY CALCULATIONS AND RANKING
C
C      EXCEPT FOR NUMBER OF CYLINDERS AND COMPRESSOR SPEED,
C      EACH INPUT IS MULTIPLIED BY 1.05 AND A NEW EAELCC IS COMPUTED.
C      NEW VALUE DIVIDED BY OLD VALUE IS COST RATIO.  BECAUSE NCY MUST
C      BE INTEGER VALUE, IT IS INCREMENTED BY ONE RATHER THAN MULTIPLIED
C      BY 1.05.  FOR RPM, TWO VALUES ARE ALLOWED.  SENSITIVITY IS BASED
C      ON SWAPPING VALUE OF RPM.  COST RATIOS CALCULATED AS BEFORE.
C      TO COMPARE COST RATIOS FOR ALL DATA VALUES, EACH IS DIVIDED BY
C      1.05, EXCEPT COST RATIO FOR NUMBER OF CYLINDERS, WHICH IS DIVIDED
C      BY THE QUANTITY (NCY+1)/NCY, AND FOR RPM, WHICH IS DIVIDED BY
C      "NEW"/"OLD".
C
C      A COST RATIO WILL EXIST FOR EACH INPUT.  THE RATIOS ARE
C      RANKED BY A SORTING ROUTINE.
C
      AATEST=4.
C      AATEST SETS UP CONDITION TO SUPPRESS EDIT SCREEN WHEN IN
C       SENSITIVITY SUBROUTINE.  (EDIT SCREEN FROM SUBROUTINE
C        COMPUTE.)
      CALL COMPUTE
      OLDCST=EAETSC
C      SAVES "BASELINE" COST VALUE
C
      RATIO=(DATA(1)+1.)/DATA(1)
      DATA(1)=DATA(1)+1.
C      SPECIAL CASE FOR NCY--INCREMENT BY ONE RATHER THAN MULT BY
C       1.05.
C
      CALL COMPUTE
      DATA(1)=DATA(1)-1.
C      RETURNS DATA(1) TO ORIGINAL VALUE
```

```
      TEMP(1)=(EAETSC-OLDCST)/RATIO
C     TEMP VALUES ARE NEW-OLD $ DIVIDED BY NEW/OLD INPUT RATIO
C
      IF (DATA(2).EQ. 3600.) THEN
      RPMNEW=1800.
      RPMOLD=3600.
      ENDIF
      IF (DATA(2).EQ. 1800.) THEN
      RPMNEW=3600.
      RPMOLD=1800.
      ENDIF
      IF (DATA(2).EQ. 1200.) THEN
      RPMNEW=1800.
      RPMOLD=1200.
      ENDIF
      IF (DATA(2).EQ.  900.) THEN
      RPMNEW=1200.
      RPMOLD= 900.
      ENDIF
45    RATIO=RPMNEW/RPMOLD
      DATA(2)=RPMNEW
      CALL COMPUTE
      DATA(2)=RPMOLD
      TEMP(2)=(EAETSC-OLDCST)/RATIO
      DO 50 K=1,2
      L=K+2
      IF (TEMP(K).GT.0.) THEN
      B(K)=1.0
      ENDIF
C
      IF (TEMP(K).EQ.0.) THEN
      B(K)=0.0
      ENDIF
C
      IF (TEMP(K).LT.0.) THEN
      B(K)=-1.0
      ENDIF
C
      TEMP(K)=ABS(TEMP(K))
C
      IF (B(K).EQ.-1.0) THEN
      CALL TCSTXT (L,66,0,3,1,'-')
      ENDIF
C
      IF (B(K).EQ.0.0) THEN
      CALL TCSTXT (L,66,0,3,1,'0')
      ENDIF
C
      IF (B(K).EQ.1.0) THEN
      CALL TCSTXT (L,66,0,3,1,'+')
```

```
      ENDIF
50    CONTINUE
C
      DO 42 K=3,21
C     INCREASE INPUT DATA VALUES BY 5%
      DATA(K)=DATA(K)*1.05
      CALL COMPUTE
      TEMP(K)=(EAETSC-OLDCST)/1.05
C
      IF (TEMP(K).LT.0.0) THEN
      B(K)=-1.0
      ENDIF
C
      IF (TEMP(K).EQ.0.0) THEN
      B(K)=0.0
      ENDIF
C
      IF (TEMP(K).GT.0.0) THEN
      B(K)=1.0
      ENDIF
C
      TEMP(K)=ABS(TEMP(K))
C
      DATA(K)=DATA(K)/1.05
C
      L=K+2
C
      IF (B(K).EQ.-1.0) THEN
      CALL TCSTXT (L,66,0,3,1,'-')
      ENDIF
C
      IF (B(K).EQ.0.0) THEN
      CALL TCSTXT (L,66,0,3,1,'0')
      ENDIF
C
      IF (B(K).EQ.1.0) THEN
      CALL TCSTXT (L,66,0,3,1,'+')
      ENDIF
C
42    CONTINUE
C
      DO 230 J=1,21
      XMAX=-1.E30
      DO 210 I=1,21
      IF(TEMP(I).GE.XMAX) XMAX=TEMP(I)
210   CONTINUE
C
      DO 220 I=1,21
      IF(TEMP(I).EQ.XMAX) THEN
                          IRANK(I)=J
```

```
                        TEMP(I)=-1.E30
                        ENDIF
220   CONTINUE
230   CONTINUE
C
      AATEST=1.
      DO 240 I=1,21
      II=I+2
240   CALL TCSIN2(II,50,0,4,IRANK(I))
C
      CALL TCSKEY(ISCAN,IASC,NCHAR)
      BBTEST=4.
C     ***SETS TEST COND. SO THAT COMPARISON SIGNS FROM SUBROUTINE
C        EDIT DO NOT DEFAULT TO ZERO WHEN NOT EDITING.
      RETURN
      END
C
C
      SUBROUTINE TCSIN2(IROW,ICOL,IATTRB,IATTRF,NO)
      CHARACTER*2 N1
      WRITE(N1,1) NO
1     FORMAT(I2)
      CALL TCSTXT(IROW,ICOL,IATTRB,IATTRF,2,N1)
      RETURN
      END
C
      SUBROUTINE XOPT(EAETSC)
C
C     THIS PROGRAM PERFORMS THE MATHEMATICAL CALCULATIONS AS DESCRIBED
C     IN CHAPTER 10 OF PROCUREMENT AND INVENTORY SYSTEMS ANALYSIS,
C     REPAIRABLE ITEM INVENTORY SYSTEMS, BY BANKS AND FABRYCKY.
C     BASED UPON SOURCE DEPENDENT DATA READ FROM DEP.DAT AND INDEPENDENT
C     SOURCE DATA READ FROM INDEP.DAT THE DESIGN VARIABLES ARE
C     OPTIMIZED AND THE EXPECTED ANNUAL EQUIVALENT TOTAL SYSTEM COST
C     (EAETSC) IS DISPLAYED.
C
C ]                                                               ^
      COMMON/INDEP/ IDEMAND, XINT, SHORT, RFAC, RFAOP
C ]                                                               ^
C ]   IDEMAND = DEMAND,  Integral Number of Units                 ^
C ]   XINT = INTerst rate                                         ^
C ]   SHORT = SHORTage penalty.                                   ^
C ]   RFAC = annual Repair FACilty cost.                          ^
C ]   RFAOP = Repair Facilty Annual OPerating cost.               ^
C ]                                                               ^
      COMMON/DEP/ACQ,SALV,IDLIFE,AOPFU,AOPLA,AOPPM,AOPOTH,XMTBF(20)
     1 ,XMTTR(20),ES
C ]                                                               ^
C ]   ACQ = ACQuisition cost.                                     ^
C ]   SALV = SALVage value at end of design life.                 ^
```

```
C ]    IDLIFE = Design LIFE of the equipment (YEARS)                     ^
C ]    AOPFU = Annual OPerating cost (FUel).                             ^
C ]    AOPLA = Annual OPerating cost (LAbor).                            ^
C ]    AOPPM = Annual OPerating cost (Preventive Maintenance).           ^
C ]    AOPOTH = Annual OPerating cost (OTHer).                           ^
C ]    XMTBF = array containing Mean Time Between Failures. (Hours)      ^
C ]    XMTTR = array containing Mean Time To Repair the equipment.       ^
C ]                                          (Hours)                     ^
C ]   MTTR and MTBF are defined as arrays of length 20.  If the          ^
C ]   design life of the equipment in question is greater than this     ^
C ]   (if IDLIFE > 20), these arrays must be increased in length.        ^
C ]                                                                      ^
C ]   ***CHECK CALC. OF IDLIFE TO SEE IF LIMITED TO 20****               ^
C
C     Array FACT is used to pre-calculate factorials. If the decision
C     variable N (the number of units to deploy) exceeds 50, the array
C     must be increased.
C
      REAL*4 FACT
C
      COMMON /ABLK/ FACT(33)
      COMMON/REPS/JM,JNN,JN
C
      DATA ICLLD/0/
C
C     PRE-CALCULATE THE NECESSARY FACTORIALS IF NECESSARY
C
      IF(ICLLD.EQ.0) THEN
         FACT(0) = 1.0
         DO 10 I=1,33
            FACT(I) = FACT(I-1)*FLOAT(I)
10       CONTINUE
         ICLLD=1
      ENDIF
C
C     TO CHECK IF DEMAND IN ACCEPTABLE RANGE
C
      IF (IDEMAND.LT.4.OR.IDEMAND.GT.20) THEN
         EAETSC = 0.0
         RETURN
      END IF
C
C     THE SEARCH FOR THE OPTIMUM BEGINS HERE.
C     THE OUTSIDE LOOP WILL BE N (THE NUMBER OF UNITS TO DEPLOY).
C     ASSIGN N THE STATED IDEMAND AS AN INITIAL VALUE.
C

      N = IDEMAND
C
      CALL OPTNN(N,TRY)
```

```
C
C     SUBROUNTINE OPTNN WILL FIND THE BEST VALUE OF NN (THE RETIREMENT
C     AGE OF THE EQUIPMENT GIVEN N AND RETURN THE CORRESPONDING EAETSC.
C
25    SUBOPT = TRY
C
C     INCREMENT N AND FIND THE BEST EAETSC FOR THIS VALUE OF N.
C
      N = N+1
C
      CALL OPTNN(N,TRY)
C
      IF(TRY.LE.SUBOPT) GO TO 25
C
C     THE SEARCH WILL CONTINUE AS LONG AS THE INCREASE IN N PROVIDES
C     BETTER EAETSC'S.
C
C     IF IT DOES NOT, THE OPTIMUM HAS BEEN FOUND.
C
      JN = N-1
      CALL OPTNN(JN,EAETSC)
      RETURN
      END
C
C
      SUBROUTINE OPTNN(N,TRY)
C
C     Subroutine OPTNN will search for the best value of NN (the
C     retirement age of the equipment) for the given value of N.
C
      REAL MOPT
      REAL*4 FACT
      COMMON /ABLK/ FACT(33)
      COMMON/INDEP/ IDEMAND, XINT, SHORT, RFAC, RFAOP
      COMMON/DEP/ACQ,SALV,IDLIFE,AOPFU,AOPLA,AOPPM,AOPOTH,XMTBF(20)
     1 ,XMTTR(20),ES
      COMMON/REPS/IM,INN,IN
C
C     ASSIGN NN THE VALUE OF THE DESIGN LIFE OF THE EQUIPMENT.  THIS
WILL
C     BE DECREASED TOWARD ZERO.  (THIS COULD ALSO BE ACCOMPLISHED BY
C     ASSIGNING AN INTITAL VALUE OF ZERO AND INCREASING TOWARD IDLIFE.
C
      NN = 1
C
      CALL OPTM(NN,N,MOPT)
C
C     SUBROUTINE OPTM WILL SEARCH FOR THE BEST VALUE OF M (THE NUMBER
C     OF REPAIR CHANNELS) AND RETURN EAETSC FOR THE GIVEN VALUES OF
C     NN AND N.
```

```
C
20    TRY = MOPT
C
C     NN SHOULD EXCEED DESIGN LIFE.
C
      NN = NN+1

      IF(NN.GT.IDLIFE) GO TO 25
C
C     INCREASE NN AND FIND THE BEST EAETSC FOR THIS VALUE OF NN AND N.
C
      CALL OPIM(NN,N,MOPT)
C
      IF(MOPT.LE.TRY) GO TO 20
C
C     THE SEARCH WILL CONTINUE AS LONG AS THE CHANGE IN NN PROVIDES
C     BETTER EAETSC'S.
C
25    INN=NN-1
      RETURN
      END
C
C
      SUBROUTINE OPIM(NN,N,MOPT)
C
C     Subroutine OPIM will search for the best value of M (the number
C     of repair channels) given the values of NN and N.
C
      REAL MOPT
      COMMON/REPS/IM,INN,IN
C
C     ASSIGN M THE VALUE OF 1.
C
      M = 1
C
      TC = CALCUL(NN,N,M)
C
C     SUBROUTINE CALCUL WILL CALCULATE THE EAETSC BASED UPON THE STATED
C     DESIGN INDEPENDENT AND DEPENDENT DATA AND THE GIVEN VALUES OF NN,
C     N AND M.
C

20    MOPT = TC
C
C     INCREMENT THE VALUE OF M.
C
      M = M+1
C
      IF(M.GT.N) GO TO 30
C
```

```
C     THE VALUE OF M SHOULD NOT EXCEED THE VALUE OF N.
C
      TC = CALCUL(NN,N,M)
C
      IF(TC.LE.MOPT) GO TO 20
C
C     THE SEARCH WILL CONTINUE AS LONG AS THE INCREASE IN M PROVIDES
C     BETTER EAETSC'S.
C
30    IM = M-1
      RETURN
      END
C
C
      FUNCTION CALCUL(NN,N,M)
C
C     Subroutine CALCUL calculates the EAETSC for the provided set of
C     design independent and dependent data and the given values of NN,
C     N and M.  These calculations follow those set forth in chapter 10
C     of Procurement and Inventory Analysis, by Banks and Fabrycky.
C     Comments will include text equation numbers where possible.
C
      REAL P(50)
      REAL*4 FACT, C(50), XX
C
      COMMON /ABLK/ FACT(33)
      COMMON/INDEP/ IDEMAND, XINT, SHORT, RFAC, RFAOP
      COMMON/DEP/ACQ,SALV,IDLIFE,AOPFU,AOPLA,AOPPM,AOPOTH,XMTBF(20)
     1 ,XMTTR(20),ES
      COMMON/RESULTS/RESULTS(14)
C
C     B = Book value
C     equ. (10.9)
C
      B = ACQ-FLOAT(NN)*((ACQ-SALV)/IDLIFE)
C
C     AEIC = Annual Equivalent Item Cost per unit
C     equ. (10.8)
C
      AAA = XINT*((1.0+XINT)**FLOAT(NN))
      BBB = ((1.0+XINT)**FLOAT(NN))-1.0
      AEIC = ACQ*(AAA/BBB)-B*(XINT/BBB)
C
C     XIC = annual equivalent Item Cost
C
      XIC = AEIC*FLOAT(N)
C
C     AERCC = Annual Equivlent Repair Channel Cost per channel
C
      AERCC = RFAC+RFAOP
```

```
C
C     RC = annual equivalent Repair Cost
C
      RC = AERCC*FLOAT(M)
C
C     XF and XR will accumulate the design's XMTBF and XMTTR.
C
      XF = 0
      XR = 0
C
      DO 25 I=1,NN
      XF = XF+XMTBF(I)
      XR = XR+XMTTR(I)
25    CONTINUE
C
C     XBARF and XBARR will be the average XMTBF and XMTTR.
C     equ. (10.10 and 10.11)
C
      XBARF = XF/FLOAT(NN)
      XBARR = XR/FLOAT(NN)
C
C     GM is the failure rate divided by the repair rate.
C
      GM = XBARR/XBARF
C
C     Perform equ. (10.14)
C
      DO 50 J=1,M
      C(J) = (FACT(N)/(FACT(N-J)*FACT(J)))*(GM**J)
50    CONTINUE
C
      DO 75 K=M+1,N
C      WRITE(*,*) ' M, N ',M,N
      XX = FLOAT(M)**(K-M)
C      WRITE(*,*) ' XX ',XX
      C(K) = (FACT(N)/(FACT(N-K)*FACT(M)*XX))*(GM**K)
      LLK=N-K
C      WRITE(*,*) ' N-K, FACT(N-K) ',LLK, FACT(LLK)
C      WRITE(*,*) ' C(K)', C(K)
75    CONTINUE
C
C     Calculate the sum of the array C just found.
C     C(0) = 1.0  The summation will start at 1.0
C
      SUMC = 1.0
      DO 100 II=1,N
      SUMC = SUMC+C(II)
100   CONTINUE
C
C     equ. (10.13)
```

```
C
      P0 = 1.0/SUMC
C
C     Compute the array P.
C
      DO 125 JJ=1,N
      P(JJ) = P0*C(JJ)
125   CONTINUE
C
C     ES = Expected number of items Short
C     equ. (10.15)
C
      ES = 0
C
      DO 150 JK=1,IDEMAND
      ES = ES+(JK*P(N-IDEMAND+JK))
150   CONTINUE
C
C     SC = annual Shortage Cost
C     equ. (10.16)
C     SHORTAGE COST IS SHORTAGE PENALTY IN $ PER BTU/MIN
C      MULT. BY NUMBER OF UNITS EXPECTED "DOWN" AT ANY TIME, MULT.
C      BY CAPACITY OF ONE UNIT.
      SC = SHORT*ES*RESULTS(1)
C
C     OC = annual Operating Cost
C
      OC =(AOPFU+AOPLA+AOPPM+AOPOTH)*(N-ES)
C
C     TC = annual equivalent Total Cost
C     equ. (10.1)
C
      CALCUL = XIC+RC+SC+OC
      RETURN
      END
C
C
      SUBROUTINE CONSTRAINTS
      LOGICAL ESC
      CHARACTER*33 LABEL33(21)
      DATA LABEL33/'                  Number of Cylinders',
     1 '                  Crank Speed, RPM','                  Piston Diamete
     2r, In','   Maximum Cylinder Volume, Cu In',
     3 ' Cylinder Clearance Volume, Cu In',
     4 'Condenser-Mean Temperature, Deg F',
     5 ' Condenser, Inlet Air Temp, Deg F',
     6 '  Condenser, Exit Air Temp, Deg F',
     7 'Evaporator, Inlet Air Temp, Deg F',
     8 ' Evaporator, Exit Air Temp, Deg F',
     9 ' Condenser      Fin Width, In (C)',
```

```
     * ' Condenser  Fin Thickness, In (C)',
     1 ' Condenser      Fin Pitch, In (C)',
     2 ' Condenser   Tube Radius,  In (C)',
     3 ' Condenser  Air Velocity, fpm (C)',
     4 ' Evaporator     Fin Width, In (E)',
     5 ' Evaporator Fin Thickness, In (E)',
     6 ' Evaporator     Fin Pitch, In (E)',
     7 ' Evaporator   Tube Radius, In (E)',
     8 ' Evaporator Air Velocity, fpm (E)',
     9 'Evaporator-Mean Temperature,Deg F'/
C
      CALL TCSTXT(0,17,0,2,38,'DESIGN FEATURE INPUT VALUE CONSTRAINTS')
      CALL TCSTXT(1,8,0,2,14,'DESIGN FEATURE')
      CALL TCSTXT(1,42,0,2,10,'CONSTRAINT')
10    DO 11 N=1,21
      NN=N+1
      CALL TCSTXT(NN,1,0,3,33,LABEL33(N))
11    CONTINUE
      CALL TCSTXT( 2,40,0,3, 4,'>= 1')
      CALL TCSTXT( 3,40,0,3,28,'900, 1200, 1800, or 3600 RPM')
      CALL TCSTXT( 4,40,0,3, 4,'>= 1')
      CALL TCSTXT( 5,40,0,3, 6,'>= 1.5')
      CALL TCSTXT( 6,40,0,3, 4,'>= 1')
      CALL TCSTXT( 7,40,0,3,24,'> EVAPORATOR TEMPERATURE')
      CALL TCSTXT( 8,40,0,3,35,'< CONDENSER EXH. AIR TEMP. & >=-457')
      CALL TCSTXT( 9,40,0,3,33,'> CONDENSER INLET AIR TEMPERATURE')
      CALL TCSTXT(10,40,0,3,33,'> EVAPORATOR EXH. AIR TEMPERATURE')
      CALL TCSTXT(11,40,0,3, 7,'>= -457')
      CALL TCSTXT(12,40,0,3,25,'> 2*CONDENSER TUBE RADIUS')
      CALL TCSTXT(13,40,0,3, 6,'>= .02')
      CALL TCSTXT(14,40,0,3,30,'>= CONDENSER FIN THICKNESS +.1')
      CALL TCSTXT(15,40,0,3, 7,'>= .125')
      CALL TCSTXT(16,40,0,3, 3,'> 0')
      CALL TCSTXT(17,40,0,3,26,'> 2*EVAPORATOR TUBE RADIUS')
      CALL TCSTXT(18,40,0,3, 6,'>= .02')
      CALL TCSTXT(19,40,0,3,28,'>= EVAPORATOR FIN PITCH + .1')
      CALL TCSTXT(20,40,0,3, 7,'>= .125')
      CALL TCSTXT(21,40,0,3, 3,'> 0')
      CALL TCSTXT(22,40,0,3, 6,'> -457')
C
      CALL TCSTXT(23,18,0,2,32,'HIT F1, F2 OR ESCAPE TO CONTINUE')
13    CALL GETFKEY(IFKEY,1,2,ESC)
      IF (ESC) GO TO 15
      IF (IFKEY.EQ.2) GO TO 15
      IF (IFKEY.EQ.1) GO TO 15
      GO TO 13
15    CONTINUE
      END
C
      SUBROUTINE BANNER
```

```
      LOGICAL ESC
      CALL TCSMOD(0)
      DO 100 N=5,19
      CALL TCSTXT( N,10,0,192,1,': ')
      CALL TCSTXT( N,69,0,192,1,': ')
100   CONTINUE
      DO 110 N=11,68
      CALL TCSTXT( 5,N,0,192,1,'M')
      CALL TCSTXT(19,N,0,192,1,'M')
110   CONTINUE
      CALL TCSTXT(19,10,0,192,1,'H')
      CALL TCSTXT( 5,10,0,192,1,'I')
      CALL TCSTXT(19,69,0,192,1,'<')
      CALL TCSTXT( 5,69,0,192,1,';')
      CALL TCSTXT( 7,33,0,9,13,'REFRIGERATION')
      CALL TCSTXT( 8,36,0,9,6,'SYSTEM')
      CALL TCSTXT( 9,35,0,9,8,'DESIGNER')
      CALL TCSTXT(14,35,0,9,8,'VPI & SU')
      CALL TCSTXT(15,37,0,9,4,'1989')
      CALL TCSTXT(17,15,0,9,11,'VERSION 2.0')
      CALL TCSTXT(18,40,0,9,22,'HIT ESCAPE TO CONTINUE')
C
115   CALL GETFKEY(IFKEY,1,2,ESC)
      IF (ESC) GO TO 125
      IF (IFKEY.EQ.2) GO TO 120
      IF (IFKEY.EQ.1) GO TO 120
120   GO TO 115
125   CONTINUE
C
C     ERASE SCREEN BEFORE LEAVING SUBROUTINE
C
      DO 130 N=0,23
      DO 140 M=0,79
      CALL TCSTXT(N,M,0,9,1,' ')
140   CONTINUE
      CALL TCSTXT(N,M,0,9,1,' ')
130   CONTINUE
      END
```