

A SOFTWARE PACKAGE TO ASSIST IN CONDUCTING
ENERGY AUDITS IN MANUFACTURING PLANTS

[A u d i t s o f t]

CREATIVE COMPONENT

Submitted by

KRIS SREEDHARAN

Submitted to

DR. WAYNE C. TURNER
Regents Professor
Industrial Engineering and Management
Oklahoma State University

Summer, 1992

ABSTRACT

This report presents the development and User's Manual for AuditSoft, a software package to assist in conducting energy audits in manufacturing plants. The report starts with an introduction which describes the usefulness of a package like AuditSoft for the EADC (Energy Analysis and Diagnostic Center), and the salient features of AuditSoft.

The User's Manual of AuditSoft included in this report is elaborate and covers details of installation, program files, menus, sample sessions, formulas and methods of calculation used in the various ECO's (Energy Conservation Opportunities), and details on using AuditSoft for EADC audits.

The appendices include sample reports produced using AuditSoft and complete listing of all program files of AuditSoft.

The disk that accompanies this report contains all the program files required to install AuditSoft.

CONTENTS

Abstract	i
1. Introduction	1
2. Description of AuditSoft & User's Manual	3
2.1 What is AuditSoft ?	3
2.2 Starting AuditSoft	4
2.3 AuditSoft Menus	5
2.4 AuditSoft Sample Session	10
2.5 Formulas used in AuditSoft ECO's	19
2.6 Using AuditSoft for EADC audits	27
2.7 AuditSoft Program Files and Directories	27
2.8 Installing AuditSoft	28
3. Conclusion	30
References	31
Appendix A:	
AuditSoft Sample Outputs	32
Appendix B:	
AuditSoft Program Listing	42
Appendix C:	
Portable Printer Information	118

1. INTRODUCTION

The Energy Analysis and Diagnostic Center (EADC) of OSU has been successfully conducting energy audits in manufacturing plants in an around Oklahoma for many years. As a Research Assistant for this program, I realized the need for a software package to automate many of the Energy Conservation Opportunities (ECO's) to assist in conducting the energy audits for the following important reasons:

- EADC plant energy audits are conducted within a day. Due to time constraints, it is virtually impossible to do initial feasibility calculations manually on site to check whether pursuing a particular energy conservation opportunity is worthwhile or not from economic standpoint.
- Many of the energy conservation opportunities are standardized. Time spent in manually calculating savings and payback for these opportunities can be saved and effectively utilized for exploring other energy conservation opportunities that are specific for the particular plant.
- Data collection becomes more streamlined. Chances of missing data are minimized.
- With a portable Personal Computer, a portable printer, and appropriate software, the whole process of energy audit could be made more efficient.
- Most of the data required from catalogs and reference books could be stored in databases. Referring to these data

becomes easier.

- Filing of audit data becomes more streamlined. In most cases, excessive paper-work could be avoided.
- Results of most of the ECO's of the audit could be presented to the client immediately after the audit.
- Preparation of final audit report takes less time.

This report presents AuditSoft, a software package developed to assist in conducting energy audits in manufacturing plants. Even though AuditSoft is tailored to follow EADC formats, it could also be used for general plant energy audits.

AuditSoft :

- Is totally menu-driven and user-friendly
- Incorporates ECO's in the areas of Lighting, Boilers, Compressors, and Electric Motors.
- Has provision to produce output text files that are compatible with word-processors so that the outputs could be directly incorporated into audit reports.
- Has provisions to expand the software to include more areas later on.

This report includes complete description of AuditSoft, an exhaustive user's manual, sample outputs, program listing, and logistics of implementing AuditSoft.

2. USER'S MANUAL & DESCRIPTION OF SOFTWARE

2.1 What is AuditSoft ?

AuditSoft is a software to assist in conducting energy audits in manufacturing plants. AuditSoft is tailored to follow the EADC (Energy Analysis and Diagnostic Center) report formats but this software could also be used for general plant energy auditing. AuditSoft is very user-friendly and reports created using this software could be printed directly or stored in text-file format (ASCII) for further editing using word-processors. AuditSoft has built-in databases which prompt the user with default values for most of the choices, with the option for the user to change it. AuditSoft can produce printouts of Energy Conservation Opportunities (ECO's) at the audit site itself.

AuditSoft has been developed using Foxpro2, a powerful relational database management system, to run on IBM-PC compatible computers.

2.2 Starting AuditSoft

At the C:> prompt type:

```
AUDITSOFT    <Press Enter>
```

This brings up a command window. Type:

```
DO AUDITSOFT <Press Enter>
```

This brings up AuditSoft logo. Press any key to continue.

For more detailed information on the AuditSoft files and directories, see the section "AuditSoft Program Files".

2.3 AuditSoft Menus

After the "DO AUDITSOFT" command, the following logo appears:

A SOFTWARE PACKAGE TO ASSIST IN CONDUCTING ENERGY AUDITS
IN MANUFACTURING PLANTS

>>>>> A u d i t S o f t <<<<<

CREATIVE COMPONENT

Submitted to:
DR. WAYNE C. TURNER
Regents Professor
Industrial Engineering & Management
Oklahoma State University

Developed by:
KRIS SREEDHARAN

Press any key to continue ...

Press any key to continue and the following MAIN MENU
appears:

MAIN MENU

LIGHTING
COMPRESSORS
ELECTRIC MOTORS
BOILERS

QUIT

AuditSoft deals with the four areas of energy auditing mentioned in the Main Menu.

To choose any of the options, use the <UP ARROW> or <DOWN ARROW> keys. The current option will be highlighted. Press Return to choose the current option.

If you want to quit to the previous command window, press <Esc>.

(Pressing <Esc> in any menu screen will make the previous menu appear.)

Choosing "LIGHTING" option makes the following window appear:

```

      LIGHTING ECOS
-----
RETROFIT EXISTING STANDARD FL LAMPS WITH EE FL LAMPS
RETROFIT FL F40T12 4 LAMP FIXT. WITH EFFICIENT 2 LAMP FIXT.

      QUIT

```

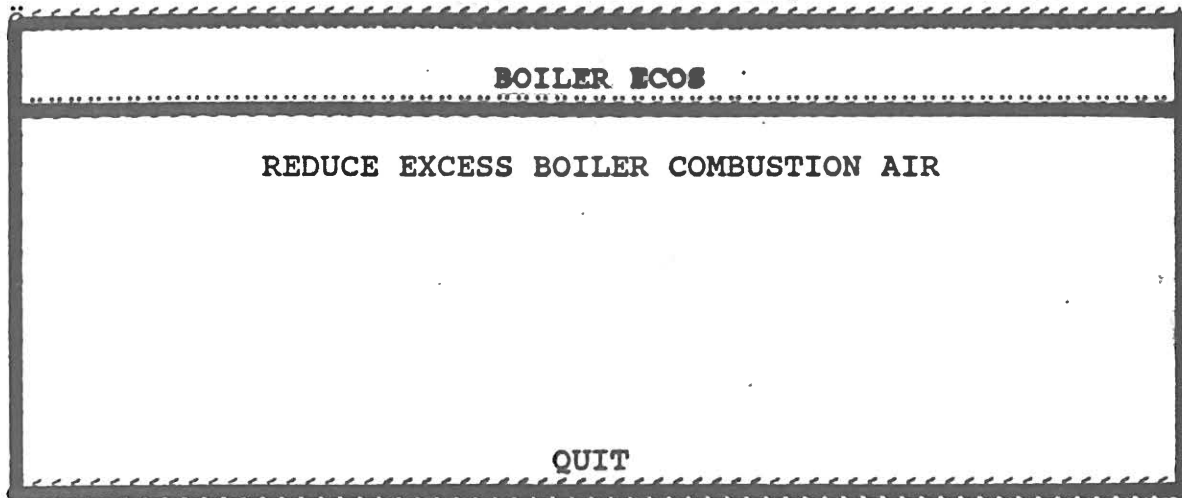
Choosing "COMPRESSORS" option makes the following window to appear:

COMPRESSOR ECOS
ELIMINATE COMPRESSED AIR LEAKS REDUCE COMPRESSED AIR PRESSURE TO MINIMUM REQUIRED
QUIT

Choosing "ELECTRIC MOTORS" option makes the following window to appear:

ELECTRIC MOTOR ECOS
USE VARIABLE FREQUENCY DRIVES USE ENERGY EFFICIENT BELTS USE SYNCHRONOUS BELTS WITH SPROCKET DRIVES
QUIT

Choosing "BOILERS" option makes the following window to appear:



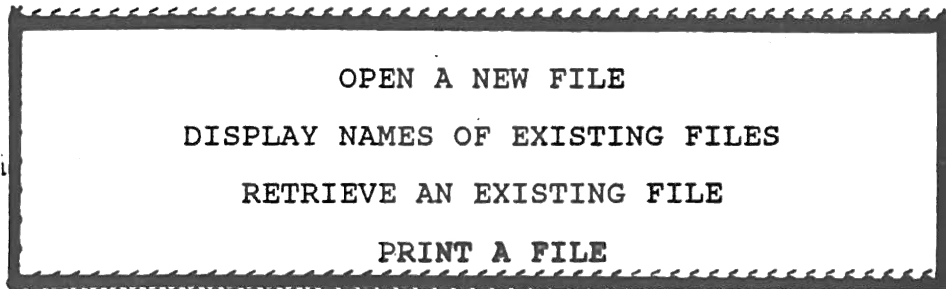
The following description is valid for the above-mentioned four menu screens - LIGHTING, COMPRESSORS, ELECTRIC MOTORS, and BOILERS:

To choose any of the options, use the <UP ARROW> or <DOWN ARROW> keys. The current option will be highlighted.

Press Return to choose the current option.

If you want to quit to the previous menu, press <Esc>.

If you choose an option, the following window will appear:



Choosing "OPEN A NEW FILE" will make the ECO data screen to appear.

Choosing "DISPLAY EXISTING FILES" will bring up a window which displays all the data files of the particular ECO, stored in the FILES directory.

Choosing "RETRIEVE AN EXISTING FILE" will bring up a window to enter the name of the file to be retrieved.

Choosing "PRINT A FILE" will bring up a window which gives the user option to print the file to printer or to a text-file.

More information on entering data, saving a data file, retrieving an existing file, and printing files are explained in detail in the section "AuditSoft Sample Session".

2.4 AuditSoft Sample Session

In this section a sample session on AuditSoft is presented.

At c:> prompt type: auditsoft <Press return>

In the command window type: do auditsoft

AuditSoft logo appears.

Press any key to continue.

Main Menu appears.

Choose "Lighting" option by moving the arrow keys.

Press return.

The following window appears.

LIGHTING ECOS
RETROFIT EXISTING STANDARD FL LAMPS WITH EE FL LAMPS
RETROFIT FL F40T12 4 LAMP FIXT. WITH EFFICIENT 2 LAMP FIXT.
QUIT

Choose "RETROFIT EXISTING STANDARD FL LAMPS WITH EE FL LAMPS".

The following window appears.

```
OPEN A NEW FILE
DISPLAY NAMES OF EXISTING FILES
RETRIEVE AN EXISTING FILE
PRINT A FILE
```

Choose "OPEN A NEW FILE" to enter fresh data.

The following window appears.

```
F40T12/CW
F96T12/CW
F96T12/CW/HO
F96T12/CW/VHO
CHOOSE EXISTING LAMP TYPE & PRESS RETURN
```

Choose "F96T12/CW/HO" and press return.

The following screen appears.

ECO: REPLACE STANDARD FLUORESCENT LAMPS WITH ENERGY EFFICIENT LAMPS
AUDIT DATA

Existing lamp type: F96T12/CW/HO	File name: L1	.MEM	Date: / /
Audit code:	Electricity consumption cost: \$	/kWh	Demand cost: \$ /kW-month
Number of lamps:	Operating hours per year:		

The user is prompted to enter data for Audit Code.

Type: Unit Parts

Next is the File name.

As you can see, after File name there is the default "L1" and ".MEM". "L1" stands for "Lighting ECO #1" and ".MEM" stands for "memory data file". This convention is used for all ECO's in AuditSoft. The second Lighting ECO will have the File name prefix as "L2", the first Boiler ECO will have the prefix as "B1", and so on. All data files are stored with the extension ".MEM". This convention helps to distinguish and sort data files according to ECO's.

For File name type in: unit

So now this data file, if saved, will be saved in the name "L1UNIT.MEM", where "L1" and ".MEM" are default.

Type in the following other data.

Date: 07/22/92

Electricity consumption cost: \$ 0.03431/kWh

Demand cost: \$5.3/kW-month

Number of lamps: 120

Operating hours per year: 5000

At this stage a blinking message appears the bottom of the screen: "ANY CORRECTIONS? (Y/N)".

Type in "Y" if you need to make any corrections to the data you have just entered.

Type in "N" to get the next window which is as follows.

LAMP DATA			
LAMP WATTAGE:	Present:	110 W	Proposed: 95 W
LAMP COST:	Present:	\$ 5.49	Proposed: \$ 6.45
LAMP LIFE:	Present:	12000 hrs	Proposed: 12000 hrs
LAMP OUTPUT:	Present:	9200 lumens	Proposed: 8300 lumens

In the above window "LAMP DATA" from the built-in database of AuditSoft is displayed. The user has the option now to change any data according to his requirements. If any data is changed, the new data will be used in the calculations by AuditSoft.

Press "Esc" if you do not want change any data.

At this stage a blinking message appears the bottom of the screen: "ANY CHANGES? (Y/N)".

Type in "Y" if you need to make any changes to the data.

Type in "N" to get the next window which is as follows.

ECO: REPLACE STANDARD FLUORESCENT LAMPS WITH ENERGY EFFICIENT LAMPS

```
..... AUDIT DATA .....
```

Existing lamp type:	F96T12/CW/HO	File name:	L1 unit .MEM	Date:	07/22/92
Audit code:	Unit Parts	Electricity consumption cost:	\$ 0.03431/kWh	Demand cost:	\$ 5.3 /kW-month
Number of lamps:	120	Operating hours per year:	5000		

```
.....
```

```
..... CALCULATIONS .....
```

Lamp wattage ->	Present:	110 W	Proposed:	95 W
Lamp cost ->	Present:	\$ 5.49	Proposed:	\$ 6.45
Lamp life ->	Present:	12000 hrs	Proposed:	12000 hrs
Lamp output ->	Present:	9200 lumens	Proposed:	8300 lumens

Energy savings:	9000 kWh/yr	30.71 MMBtu/yr
Dollar savings:	\$ 423 /yr	
Implementation cost:	\$ 115	(Incremental-replace as and when they fail)
Simple payback:	0.2 years	

```
.....
```

SAVE FILE? (Y/N)

As you can see AuditSoft has done the calculations and has come up with the Energy Savings in kWh/yr and MMBtu/yr, Dollar Savings, Implementation Cost, and Simple Payback.

At this stage a blinking message appears the bottom of the screen: "SAVE FILE? (Y/N)".

Type in "Y" if you need to save the file.

If you type in "Y", all the data that you entered will be saved in the file "L1UNIT.MEM".

If a file by name "L1UNIT.MEM" already existed in the FILES

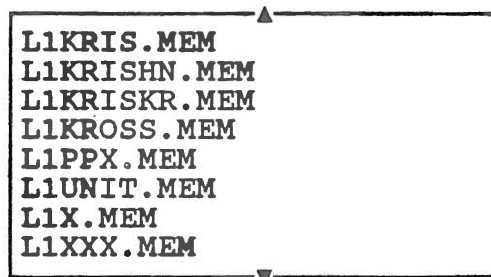
directory, AuditSoft comes up with the message that the file already exists and asks whether you want to overwrite it or not, to protect your data files.

After saving the file we are back in the following menu.

At any stage if you want to go back to the previous menu press "Esc."

Caution: Do not press "Esc" more than once at a time. This could abort the program and you might lose all the data that you have entered.

Now if you want to display the data files in the FILES directory choose "DISPLAY NAME OF EXISTING FILES" by moving the highlighting bar using the arrow keys and press return. The following window appears.



```
L1KRIS.MEM
L1KRISHN.MEM
L1KRISKR.MEM
L1KROSS.MEM
L1PPX.MEM
L1UNIT.MEM
L1X.MEM
L1XXX.MEM
```

Use the arrow keys to move the highlighting bar to the bottom of the list of files.

As you can see, names of only the data files with the "L1" prefix are displayed. (If you had chosen the first ECO of

the Boiler section, all the data files with the prefix "B1" would have been displayed.)

Press return to exit this window.

We are back to the previous window.

OPEN A NEW FILE
DISPLAY NAMES OF EXISTING FILES
RETRIEVE AN EXISTING FILE
PRINT A FILE

To retrieve an existing file choose "RETRIEVE AN EXISTING FILE" and press return.

The following window appears.

TYPE IN NAME OF EXISTING FILE: L1 .MEM

Type in name of file to be retrieved. If you enter the name of a file that does not exist AuditSoft comes up with the message that the file does not exist.

If you type in the name of a file that exists AuditSoft comes up with the Data screen with all the stored data. AuditSoft will let you make changes to the existing data if you want to. You could save the data again in the same file name or another file name.

To print a file choose the option "PRINT A FILE" and press return.

The following window appears.

TYPE IN NAME OF FILE TO BE PRINTED: L1	.MEM
--	------

Type in name of data file to be printed.

If you enter the name of a file that does not exist AuditSoft comes up with the message that the file does not exist.

Choose "PRINT TO PRINTER" to print the report directly.

If the printer is not ready AuditSoft comes up with the message that the printer is not ready and prompt the user to try again.

Choose "PRINT TO filename.txt" to print the report to a text file of the name "L1____.TXT".

This text file could be edited using Professional Write or any other Word processor to suit the user's requirements, later on.

2.5 Formulas used in AuditSoft ECO's

(I). Lighting

ECO (L1). Retrofit existing standard fluorescent lamps with energy efficient fluorescent lamps

Demand savings (kw)

$$= (\text{Number of lamps}) (\text{Watts saved}) (1 \text{ kW}/1000\text{W}) (\text{Ballast factor})$$

Consumption savings (kWh/year)

$$= (\text{Demand savings}) (\text{Operating hours per year})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings}) (3412 \text{ Btu}/\text{kWh}) (1 \text{ MMBtu}/106\text{Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings}) (\text{Electricity charge}) + (\text{Demand savings}) (\text{Demand cost}) (12 \text{ months})$$

Implementation cost (\$)

$$= (\text{Number of lamps}) (\text{Incremental cost})$$

Simple payback (years)

$$= (\text{Implementation cost}) / (\text{Annual dollar savings})$$

ECO (L2). Retrofit fluorescent F40T12 4 lamp fixture with
efficient 2 lamp fixtures

Demand savings (kw)

$$= (\text{Number of fixtures}) [(\text{Present fixture Watts} - \text{Proposed fixture watts})] / (1 \text{ kW}/1000 \text{ W})$$

Consumption savings (kWh/year)

$$= (\text{kW savings}) (\text{Operating hours})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings}) (3412 \text{ Btu/kWh}) (1 \text{ MMBtu}/106 \text{ Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings}) (\text{Electricity charge}) + (\text{Demand savings}) (\text{Demand cost}) (12 \text{ months})$$

Implementation cost (\$)

$$= [(\text{Lamp cost per lamp}) (\# \text{ of lamps per fixture}) + (\text{Reflector cost per fixture}) + (\text{Electronic ballast cost per fixture}) + (\text{Labor cost per fixture})] (\# \text{ of fixtures})$$

Simple payback (years)

$$= (\text{Implementation cost}) / (\text{Annual dollar savings})$$

(II). Compressors

ECO (C1). Eliminate compressed air leaks

Annual loss due to air leaks (kWh/yr)

$$= [(\# \text{ of } 1/8" \text{ leaks})(\text{Annual fuel wastage in kWh/yr}) + (\# \text{ of } 1/16" \text{ leaks})(\text{Annual fuel wastage in kWh/yr}) + (\# \text{ of } 1/32" \text{ leaks})(\text{Annual fuel wastage in kWh/yr})](\text{Hours air is supplied})/(8,760 \text{ hours/year})$$

Consumption savings (kWh/year)

$$= (\text{Annual energy loss due to air leaks})(\text{Estimated air leak reduction})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings})(3412 \text{ Btu/kWh})(1 \text{ MMBtu}/106\text{Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings})(\text{Electricity charge})$$

Maintenance cost (\$)

$$= (\text{Cost of repair})(\text{Annual number of leaks})$$

Net savings (\$)

$$= (\text{Annual dollar savings}) - (\text{Maintenance cost})$$

Implementation cost (\$)

$$= \text{Nil}$$

Simple payback (years)

$$= \text{Immediate}$$

ECO (C2). Reduce compressed air pressure to minimum

Demand savings (kw)

$$= (\text{Total air compressor horsepower}) (\text{Air compressor load}) \\ (\text{Approximate decrease in BHP}) (0.746 \text{ kW/HP})$$

[Approximate decrease in BHP is obtained from a database
based on the input data]

Consumption savings (kWh/year)

$$= (\text{kW savings}) (\text{Operating hours})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings}) (3412 \text{ Btu/kWh}) (1 \text{ MMBtu}/106\text{Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings}) (\text{Electricity charge}) + \\ (\text{Demand savings}) (\text{Demand cost}) (12 \text{ months})$$

Implementation cost (\$)

$$= \text{Nil}$$

Simple payback (years)

$$= \text{Immediate}$$

(III). Electric Motor

ECO (E1). Use variable frequency drives

Power consumption when VFD is not installed (kWh/HP/yr)
(constant volume air flow)

$$= (1/\text{Typical motor efficiency}) (.746 \text{ kWh/HP}) (\text{Operating hours}) (\text{Load factor})$$

Power consumption when VFD is installed (for each % rated RPM of motor) (kWh/HP/yr)

$$= (\% \text{ rated RPM})^3 (.746 \text{ kW/HP}) / (\text{Estimated system efficiency for this \% rated RPM}) (\text{Hours/year for this \% rated RPM})$$

Total power consumption when VFD is installed (for each % rated RPM of motor) (kWh/HP/yr)

$$= \text{Sum of all the power consumption values calculated for each \% rated RPM}$$

Consumption savings (kWh/year)

$$= (\text{Power consumption when VFD is not installed} - \text{Total power consumption when VFD is installed}) (\text{Total fan motor HP})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings}) (3412 \text{ Btu/kWh}) (1 \text{ MMBtu}/106\text{Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings}) (\text{Electricity charge})$$

Implementation cost (\$)

$$= (\text{Total motor fan HP}) (\text{VFD cost per unit HP})$$

Simple payback (years)

$$= (\text{Implementation cost}) / (\text{Annual dollar savings})$$

ECO (E2). Use energy efficient belts

Demand savings (kw)

$$= (\# \text{ of motors}) (\text{motor hp}) (1/\text{Motor efficiency}) (\text{Part load factor}) (\text{Percentage energy saved due to energy efficient belts}) (.746 \text{ kW/HP})$$

Consumption savings (kWh/year)

$$= (\text{kW savings}) (\text{Operating hours})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings}) (3412 \text{ Btu/kWh}) (1 \text{ MMBtu}/106\text{Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings}) (\text{Electricity charge}) + (\text{Demand savings}) (\text{Demand cost}) (12 \text{ months})$$

Implementation cost (\$)

$$= \text{Nil (Replaced as and when they fail)}$$

Simple payback (years)

$$= \text{Immediate}$$

ECO (E3). Use synchronous belts with sprocket drives

Demand savings (kw)

$$= (\# \text{ of motors}) (\text{motor hp}) (1/\text{Motor efficiency}) (\text{Part load factor}) (\text{Percentage energy saved due to energy efficient belts}) (.746 \text{ kW/HP})$$

Consumption savings (kWh/year)

$$= (\text{kW savings}) (\text{Operating hours})$$

Energy savings (MMBtu/year)

$$= (\text{Consumption savings}) (3412 \text{ Btu/kWh}) (1 \text{ MMBtu}/106\text{Btu})$$

Annual dollar savings (\$/yr)

$$= (\text{Consumption savings}) (\text{Electricity charge}) + (\text{Demand savings}) (\text{Demand cost}) (12 \text{ months})$$

Implementation cost (\$)

$$= (\text{Number of motors}) (\text{Implementation cost per motor})$$

Simple payback (years)

$$= (\text{Implementation cost}) / (\text{Annual dollar savings})$$

(IV). Boilers

ECO (B1). Reduce excess boiler combustion air

Natural gas savings (MCF/yr)

$$= (\text{Boiler fuel consumption}) (12 \text{ mo/yr}) (\text{Expected savings})$$

[Expected savings is calculated by interpolating values
obtained from a database based on input data]

Energy savings (MMBtu/yr)

$$= (\text{Natural gas savings}) (1 \text{ MMBtu/MCF})$$

Dollar savings (\$) = (Natural gas savings) (Cost of natural gas)

Implementation cost (\$)

$$= \text{Nil (Replaced as and when they fail)}$$

Simple payback (years)

$$= \text{Immediate}$$

2.6 Using AuditSoft for EADC Audits

AuditSoft has been installed on the Zenith Laptop Computer of the Energy Program. In the appendix is included information on two portable printers. If any of these printers are procured, Auditsoft could be effectively used to make printouts of ECO's at the audit site itself. However, without a portable printer, AuditSoft could be used to collect audit data and display the ECO results on screen. The collected data could be stored on disks and printed out later.

2.7 AuditSoft Program Files and Directories

The following are the program files required to run AuditSoft:

c:\foxpro2\files

MAINMENU	FXP	522	6-20-92	4:20a
MEN_CREA	FXP	2857	7-14-92	9:16a
LITE1	DBF	652	6-14-92	4:00a
LIGHTMEN	FXP	293	7-12-92	7:01a
LITE2	FXP	11506	7-15-92	2:30a
LOGO	FXP	1591	7-18-92	8:23a
COMP	FXP	289	7-10-92	6:19a
COMP1	FXP	9602	7-15-92	3:05a
ELEC	FXP	326	7-11-92	8:03a
COMP2	FXP	9020	7-15-92	4:20a

BOIL	FXP	252	7-10-92	12:57a
COMP1_D	DBF	310	6-16-92	8:11a
LITE1	FXP	11833	7-15-92	2:26a
ELEC2	FXP	7528	7-14-92	9:05p
ELEC1	FXP	14392	7-15-92	7:37a
ELEC3	FXP	9636	7-14-92	9:05p
BOIL1_D	DBF	829	7-09-92	3:23a
BOIL1	FXP	10168	7-15-92	5:19a

These files are stored in the FILES directory within the FOXPRO2 directory.

In addition to the above files AUDITSOFT.BAT is stored in the C directory.

2.8 Installing AuditSoft

The 3 1/2 inch disk that accompanies this report contains all the AuditSoft program files. AuditSoft runs under Foxpro 2. Make a sub-directory FILES within FOXPRO2 directory by typing at the c:\foxpro2> prompt:

```
mkdir files <Press return>
```

Then type:

```
cd files <Press return>
```

At the c:\foxpro2\files> prompt type:

```
copy b:*. * (or a:*. * if the disk is in the A drive)
```

```
<Press return>
```

Then type:

```
cd\ <Press return>
```

At the C:> prompt type:

copy b:*.bat (or a:*.bat if the disk is in the A drive)

<Press return>

Installation of AuditSoft is complete.

3. CONCLUSION

AuditSoft, a software package to assist in conducting energy audits in manufacturing plants, tailored for the EADC (Energy Analysis and Diagnostic Center) of Oklahoma State University has been developed for use on IBM-PC compatible computers. AuditSoft includes ECO's (Energy Conservation Opportunities) in the areas of Lighting, Compressors, Electric Motors, and Boilers.

AuditSoft, loaded on a portable personal computer, (with a portable printer) could be effectively used to collect audit data and produce ECO reports at the audit site itself.

AuditSoft has been tested with actual data collected during energy audits and the results were found to be accurate. However, there is scope for improvement of this software by adding more modules of ECO's.

REFERENCES

1. Turner, W.C. (1982). Energy Management Handbook. John Wiley & Sons, New York, N.Y.
2. Oklahoma Energy Analysis and Diagnostic Center Audit Reports.
3. FOXPRO 2 reference manuals.

APPENDIX A
AUDITSOFT SAMPLE OUTPUTS

Audit code: American candy Date: 09/09/92
File name: LICANDY

ECO: REPLACE STANDARD FLUORESCENT LAMPS WITH
ENERGY EFFICIENT FLUORESCENT LAMPS IN EXISTING FIXTURES

RECOMMENDED ACTION

Energy-efficient (EE) fluorescent lamps consume less energy than standard lamps while giving nearly the same light levels. They may cost a little more, but the additional expense is recovered through energy savings. These energy-efficient lamps do not require fixture modification or ballast upgrade. We recommend that you replace the standard fluorescent lamps with energy-efficient lamps as the standard lamps fail. The savings calculated above will not be realized until all existing lamps have failed.

DATA

Existing lamp type: F96T12/CW
Number of lamps: 120
Operating hours per year: 4000 hours/yr
Electricity consumption cost: \$0.03431 /kWh
Electricity demand cost: \$5.3 /kW-month

Existing lamp data:
Wattage: 75 W
Cost: \$ 4.24 /lamp
Life: 12000 hours
Output: 6300 lumens

Proposed lamp data:
Wattage: 60 W
Cost: \$ 5.07 /lamp
Life: 12000 hours
Output: 5600 lumens

CALCULATIONS

Energy savings = 7200 kWh/yr
 = 24.57 MMBtu/yr
Dollar savings = \$ 362 /yr
Implementation cost = \$ 100
Simple payback = 0.28 years

Audit code: American candy Date: 02/02/92
File name: L2CANDY

ECO: REPLACE STANDARD FLUORESCENT F40T12 4 LAMP FIXTURES WITH
2 HIGH EFFICIENCY, HIGH INTENSITY, LONG LIFE, LAMPS,
AN ELECTRONIC BALLAST, AND A HIGH EFFICIENCY REFLECTOR

RECOMMENDED ACTION

Energy savings can be achieved by retrofitting existing fluorescent lighting fixtures. The four lamp (48 inch) fixtures currently in place can be replaced by high efficiency, high intensity, long life two lamp fixtures which will save energy while retaining the same light level. This is accomplished by adding a reflector along with the lamps in order to point the maximum amount of light in the direction needed. Each retrofit fixture will consist of an electronic ballast, a reflector, and two lamp receptacles.

DATA

Number of fixtures: 118
Operating hours per year: 2750 hours/yr
Electricity consumption cost: \$.06 /kWh
Electricity demand cost: \$1 /kW-month

Present lighting system: Fluorescent, 4F40T12 (48 inch)
Number of lamps per fixture 4
Watts per fixture 192 W
Lumens per fixture 3,050 lumens
Lamp life 20,000 hours
Lamp cost \$ 1.77 /lamp

Proposed lighting system: Fluorescent, 4F40AX35 (48 inch)
Number of lamps per fixture 2
Watts per fixture 70 W
Lumens per fixture 3,700 lumens
Lamp life 24,000 hours
Lamp cost \$ 9.88 /lamp
Reflector cost per fixture \$ 27.00
Electronic ballast cost per fixture \$ 32.50
Labor cost for retrofitting per fixture \$ 15.00

CALCULATIONS

Energy savings = 39589 kWh/yr
 = 135.08 MMBtu/yr

Dollar savings = \$ 2548 /yr

Implementation cost = \$ 11123

Simple payback = 4.40 years

Audit code: American candy Date: 09/09/92
File name: C1CANDY

ECO: ELIMINATE COMPRESSED AIR LEAKS

RECOMMENDED ACTION

Compressed air is distributed through pipelines from the compressors to the pneumatic equipment. Because of the connections in the piping and accidental holes in the tubing, air leaks are unavoidable. However, an ongoing maintenance program to repair leaks can reduce their number and make them short-lived. By fixing the air leaks energy savings can be achieved.

DATA

Compressor discharge pressure: 120 psig
Operating hours per year: 5000 hours/yr
Electricity consumption cost: \$0.03431 /kWh
Estimated air leak reduction from repair: 90 %
Number of very large leaks (1/4 inch): 1
Number of large leaks (1/8 inch): 5
Number of medium leaks (1/16 inch): 7
Number of small leaks (1/32 inch): 8

CALCULATIONS

Energy savings = 159005 kWh/yr
 = 542.53 MMBtu/yr

Dollar savings = \$ 5455 /yr

Maintenance cost = \$ 110 /yr
(Assuming repair of 11 leaks per year
at a cost of \$10/repair, after an initial effort to
repair existing leaks)

Net dollar savings = \$ 5345 /yr

Simple payback = Immediate

Audit code: KRISCROSS Date: 09/08/92
File name: C2ZZZ

ECO: REDUCE COMPRESSED AIR PRESSURE TO MINIMUM REQUIRED

RECOMMENDED ACTION

Compressed air is distributed through pipelines from the compressors to the pneumatic equipment. The discharge pressure of compressed air system could be lowered without causing operating problems. This reduction in air pressure to the minimum required level reduces the energy consumption of the electric motors driving the compressors. When implementing this ECO it should be verified that the lower discharge pressure is acceptable for all of the equipment in the plant .

DATA

Total air compressor HP:	250 HP
Air compressor load factor:	90 %
Present compressed air pressure:	125 psig
Operating hours per year:	2000 hours/yr
Electricity consumption cost:	\$.0789 /kWh
Electricity demand cost:	\$7.8 /kW-month

CALCULATIONS

Approximate percentage decrease in BHP due to reduction
in pressure of compressed air to 100 psig : 13.3 %

Energy savings	=	44648 kWh/yr
	=	152.34 MMBtu/yr
Dollar savings	= \$	5612 /yr
Implementation cost	=	Nil
Simple payback	=	Immediate

Audit code: American candy Date: 07/22/92
 File name: E1CANDY

ECO: USE VARIABLE FREQUENCY DRIVES ON SELECTED MOTORS

RECOMMENDED ACTION

Variable Frequency Drive (VFD) units work by electronically controlling fan motor speed to meet airflow demand. VFD units can be retrofitted to any existing AC motor and fan. They take only a few hours to install, have no moving parts, and require no routine maintenance. Large savings can be obtained because reduction in a fan motor's speed would reduce the power requirements by the cube of the speed reduction. For example, if the speed is reduced by 50%, the power input will be reduced by the cube of 50%, or to 12.5% of the original power.

DATA

Estimated motor efficiency: 90 %
 Estimated motor partload factor: 99 %
 Electricity consumption cost: \$.04338 /kWh

System #	Description	# of motors	Motor HP	Oper.hours hours/yr	Implemn. cost(\$) per motor
1	Blower fans	3	3	8760	600.00
2	Blower fans	2	5	8760	800.00
3					0.00
4					0.00
5					0.00
6					0.00 -

DUTY CYCLE

% Rated RPM	% Operating hours	Estimated system efficiency
100	0	82
90	3	81
80	9	80
70	21	76
60	23	70
50	23	66
40	16	64
30	5	54

CALCULATIONS

Energy savings = 97791 kWh/yr
= 333.66 MMBtu/yr
Dollar savings = \$ 4242 /yr
Implementation cost = \$ 3400
Simple payback = 0.80 years

Audit code: American candy Date: 07/22/92
 File name: E2CANDY

ECO: REPLACE STANDARD BELTS WITH ENERGY EFFICIENT BELTS

RECOMMENDED ACTION

Cog-type energy efficient belts have been shown in field tests to offer energy savings ranging from 2 to 4% due to reduced friction between the belt and the pulleys. In addition to saving energy, belt replacement costs are also reduced, due to the longer life of the energy-efficient belt. Manufacturers claim belt lives up to twice that of the standard belt. The incremental cost of the more efficient belt ranges from 20% to 80% depending on belt type. Therefore over a period of time, the actual belt cost will be less using the more efficient belt. For this reason no additional belt costs will be incurred.

DATA

Average percent energy saved with EE belts: 3 %
 Estimated motor efficiency: 90 %
 Estimated motor partload factor: 80 %
 Electricity consumption cost: \$0.03431 /kWh
 Electricity demand cost: \$5.6 /kW-month -----

System #	Description	# of motors	Motor HP	Oper. hours hours/yr ----
1	Dust collector fan	3	25	5500
2	Blower	1	5	4700
3				
4				
5				
6	-----			

CALCULATIONS

Energy savings = 8673 kWh/yr
 = 29.59 MMBtu/yr
 Dollar savings = \$ 405 /yr
 Implementation cost = Nil
 Simple payback = Immediate

Audit code: OKLAHOMA Date: 09/09/98
 File name: E3NEW

ECO: USE SYNCHRONOUS BELTS WITH SPROCKET DRIVES

RECOMMENDED ACTION

Synchronous (or toothed) belts in combination with sprockets use less power than most other drive systems and are suited for a large range of applications. This type of belt drive system weighs up to 33 percent less than the standard one and has been shown by field testing to have energy savings ranging from 5 to 7 percent. These energy savings result from a reduction in slippage, a higher power transmission efficiency, and minimum idling power losses. Manufacturers claim that the belt life of an energy-efficient synchronous belt is as much as twice that of a standard belt. The incremental cost of the energy-efficient belt ranges from 20 to 40 % depending on the belt type. Therefore over a period of time, the actual operating cost will be less by using the more efficient belt-drive system.

DATA

Average percent energy saved with
 synchronous belt system: 5 %
 Estimated motor efficiency: 90 %
 Estimated motor partload factor: 80 %
 Electricity consumption cost: \$0.03431 /kWh
 Electricity demand cost: \$5.3 /kW-month -----

System #	Description	# of motors	Motor HP	Oper. hours/yr	Implemn. cost(\$) per motor
1	Machine #12	2	20	8760	193.00
2	Machine #22	4	15	5000	175.00
3					0.00
4					0.00
5					0.00
6					0.00 -

CALCULATIONS

Energy savings = 21565 kWh/yr
 = 73.58 MMBtu/yr
 Dollar savings = \$ 951 /yr
 Implementation cost = \$ 1086
 Simple payback = 1.10 years

Audit code: American candy Date: 09/09/92
File name: B1BOIL

ECO: REDUCE EXCESS BOILER COMBUSTION AIR

RECOMMENDED ACTION

Oxygen is required for combustion of any fuel. This oxygen is obtained from atmospheric air which is 21% oxygen by volume. In theory it is possible to supply just the right amount of air containing the oxygen required for complete combustion of the fuel. In practice, however, getting the air and fuel distributed and mixed evenly is virtually impossible. For this reason an excess of combustion air is recommended to ensure that all fuel burn completely. The amount of excess air should be kept to a minimum (about 2% oxygen) to avoid exhausting heated air from the stack. We recommend that you monitor flue gas oxygen content using the fuel efficiency analyzer and adjust air intake to maintain 2% oxygen.

DATA

Natural gas cost:	\$2.78	/MCF
Boiler fuel consumption:	3487	MCF/month
Percentage Oxygen in flue gas:	9.30	%
Temperature of flue gas:	850	F

CALCULATIONS

Expected percentage fuel savings by reducing
percentage of Oxygen in flue gas to 2% : 11.50 %

Natural gas savings: 4812.00 MCF/yr

Energy savings = 4812.00 MMBtu/yr

Dollar savings = \$ 13377 /yr

Implementation cost = Nil

Simple payback = Immediate

APPENDIX B
AUDITSOFT PROGRAM LISTING

*** PROGRAM MAINMENU.PRG

```
set talk off
set status off
set scoreboard off
set message to 23 center
*set color of scheme 1 to b+/w+, , ,gr+/b+,rg*/b+
```

```
do men_crea
do logo
```

```
activate popup main
clear
clear all
return
```

```
procedure m_cases
do case
  case bar() = 5
    activate popup light
  case bar() = 7
    activate popup comp_p
  case bar() = 9
    activate popup elec_p
  case bar() = 11
    activate popup boil_p
  case bar() = 15
    deactivate popup
  return
endcase
```

*** PROGRAM MEN_CREA.PRG

```
public e_costp,d_costp,o_hoursp,datp,n_costp
e_costp=space(7)
d_costp=space(5)
o_hoursp=space(4)
datp=space(8)
n_costp=space(5)
```

```
define popup main from 4,20 to 20,60;
  shadow
define bar 2 of main prompt cent_str("MAIN MENU",37) skip
define bar 3 of main prompt replicate(chr(205),39) skip
define bar 5 of main prompt cent_str("LIGHTING",37);
  message "Lighting Energy Conservation Opportunities (ECOS)"
define bar 7 of main prompt cent_str("COMPRESSORS",37);
  message "Compressor Energy Conservation Opportunities
(ECOS)"
define bar 9 of main prompt cent_str("ELECTRIC MOTORS",37);
  message "Electric Motor Energy Conservation Opportunitites
(ECOS)"
define bar 11 of main prompt cent_str("BOILERS",37);
  message "Boiler Energy Conservation Opportunities (ECOS)"
define bar 15 of main prompt cent_str("QUIT",37);
  message "TERMINATE THIS PROGRAM"
```

```
define popup light from 4,10 to 20,70;
  shadow
define bar 2 of light prompt cent_str("LIGHTING ECOS",59)
skip
define bar 3 of light prompt replicate(chr(205),59) skip
define bar 5 of light prompt cent_str("RETROFIT EXISTING
STANDARD FL LAMPS WITH EE FL LAMPS",59); message ""
define bar 7 of light prompt "RETROFIT FL F40T12 4 LAMP FIXT.
WITH EFFICIENT 2 LAMP FIXT."; message ""
define bar 15 of light prompt cent_str("QUIT",59);
  message "GO BACK TO THE MAIN MENU"
```

```
define popup light1 from 10,20 to 18,65;
  prompt field lamp;
  footer "CHOOSE EXISTING LAMP TYPE & PRESS RETURN";
  shadow
```

```
define popup comp_p from 4,10 to 20,70;
  shadow
define bar 2 of comp_p prompt cent_str("COMPRESSOR ECOS",59)
skip
define bar 3 of comp_p prompt replicate(chr(205),59) skip
define bar 5 of comp_p prompt cent_str("ELIMINATE COMPRESSED
AIR LEAKS",59);
  message ""
define bar 7 of comp_p prompt cent_str("REDUCE COMPRESSED AIR
PRESSURE TO MINIMUM REQUIRED",59); message ""
```

```

define bar 15 of comp_p prompt cent_str("QUIT",59);
    message "GO BACK TO THE MAIN MENU"

define popup compl_p from 10,20 to 18,65;
    prompt field pres;
    footer "CHOOSE COMPRESSOR DISCHARGE PRESSURE";
    shadow

define popup elec_p from 4,10 to 20,70;
    shadow
define bar 2 of elec_p prompt cent_str("ELECTRIC MOTOR
ECOS",59) skip
define bar 3 of elec_p prompt replicate(chr(205),59) skip
define bar 5 of elec_p prompt cent_str("USE VARIABLE
FREQUENCY DRIVES",59);
    message ""
define bar 7 of elec_p prompt cent_str("USE ENERGY EFFICIENT
BELTS",59);
    message ""
define bar 9 of elec_p prompt cent_str("USE SYNCHRONOUS BELTS
WITH SPROCKET DRIVES",59);
    message ""
define bar 15 of elec_p prompt cent_str("QUIT",59);
    message "GO BACK TO THE MAIN MENU"

define popup boil_p from 4,10 to 20,70;
    shadow
define bar 2 of boil_p prompt cent_str("BOLIER ECOS",59) skip
define bar 3 of boil_p prompt replicate(chr(205),59) skip
define bar 5 of boil_p prompt cent_str("REDUCE EXCESS BOILER
COMBUSTION AIR",59);
    message ""
define bar 15 of boil_p prompt cent_str("QUIT",59);
    message "GO BACK TO THE MAIN MENU"

on selection popup main do m cases
on selection popup light do lightmen
on selection popup light1 do litel_1
on selection popup comp_p do comp
on selection popup compl_p do compl_1
on selection popup elec_p do elec
on selection popup boil_p do boil

define window cover from 0,0 to 24,79 none

function cent_str
parameters string,stlen
private ret_val, lpad, tpad
lpad = int((stlen-len(ltrim(trim(string))))/2)
tpad = stlen-lpad-len(ltrim(trim(string)))
ret_val = space(lpad)+ltrim(trim(string))+space(tpad)
return ret_val

```



```
***program logo.prg
```

```
set echo off
set talk off
set status off
clear
```

```
an=" "
```

```
x=17
define window wlogo1 from 1,5 to 7,75 double;
  shadow
move window wlogo1 to 18,5
activate window wlogo1
@1,1 say "      A SOFTWARE PACKAGE TO ASSIST IN CONDUCTING
ENERGY AUDITS"
@2,1 SAY "              IN MANUFACTURING PLANTS"
@4,1 SAY "              >>>>> A u d i t S o f t <<<<<<"
  do while x<>0
    if x<>0
      move window wlogo1 by -1,0
      x=x-1
    endif
  enddo
```

```
x=1
define window wlogo2 from 1,1 to 11,40 double;
  shadow
move window wlogo2 to 1,6
activate window wlogo2
@1,1 SAY "      CREATIVE COMPONENT"
@3,1  SAY "      Submitted to:"
@4,1  SAY "      DR. WAYNE C. TURNER"
@5,1  SAY "      Regents Professor"
@6,1  SAY "Industrial Engineering & Management"
@7,1  SAY "      Oklahoma State University"

  do while x<>9
    if x<>9
      move window wlogo2 by 1,3
      x=x+1
    endif
  enddo
```

```
x=1
define window wlogo3 from 1,1 to 6,25 double;
  shadow
move window wlogo3 to 1,40
activate window wlogo3
@1,1  SAY "      Developed by:"
```

```
@2,1 SAY " KRIS SREEDHARAN"
```

```
do while x<>17
  if x<>17
    move window wlogo3 by 1,-2
    x=x+1
  endif
enddo
```

```
x=1
define window wlogo4 from 1,7 to 4,39 none
activate window wlogo4
do while x<>23
  if x<>23
    move window wlogo4 by 1,2
    x=x+1
  endif
enddo
wait
```

```
release windows wlogo1,wlogo2,wlogo3,wlogo4
clear
```

```
*** program lightmen.prg  
do case  
  case bar() = 5  
    do lite1  
  case bar() = 7  
    do lite2  
  case bar() = 15 deactivate popup return  
endcase
```

```
*** program comp.prg
```

```
do case  
  case bar() = 5  
    do comp1  
  case bar() = 7  
    do comp2  
  case bar() = 15 deactivate popup return  
endcase
```

```
*** program elec.prg
```

```
do case  
  case bar() = 5  
    do elec1  
  case bar() = 7  
    do elec2  
  case bar() = 9  
    do elec3  
  case bar() = 15 deactivate popup return  
endcase
```

```
*** program boil.prg
```

```
do case  
  case bar() = 5  
    do boil1  
  case bar() = 15 deactivate popup return  
endcase
```

**** PROGRAM LITE1.PRG

```
mem_file=space(6)
m_lamp=space(20)
m_watts=0
m_nwatts=0
m_cost=0
m_ncost=0
m_life=0
m_nlife=0
m_lumens=0
m_nlumen=0
a_code=space(16)
f_name=space(6)
n_lamp=space(3)
dm_svngs=0
kw_svngs=0
mb_svngs=0
d_svngs=0
im_cost=0
p_back=0
```

```
define window cover1 from 1,1 to 6,70 double;
  shadow
define window cover2 from 1,1 to 20,68 double;
  shadow
```

```
define popup fl_lite1 from 15,15 to 24,63;
shadow
define bar 2 of fl_lite1 prompt "          OPEN A NEW
FILE"
define bar 4 of fl_lite1 prompt "          DISPLAY NAMES OF
EXISTING FILES"
define bar 6 of fl_lite1 prompt "          RETRIEVE AN
EXISTING FILE"
define bar 8 of fl_lite1 prompt "          PRINT A
FILE"
on selection popup fl_lite1 do l_lite1
```

```
define popup fm_lite1 from 8,30 to 17,55;
prompt files like l1*.mem;
shadow
on selection popup fm_lite1 do m_lite1
```

```
activate popup fl_lite1
return
```

```
procedure l_lite1
do case
  case bar()=2
    do lite1_nw
```

```

case bar()=4
  activate popup fm_litel
case bar()=6
  do litel_7
case bar()=8
  do litel_9

endcase
return

```

```

procedure m_litel
deactivate popup
return

```

```

procedure litel_nw
use litel
activate popup light1
if lastkey()<>13
  return
endif
do litel_2
do litel_4
do litel_4a
do litel_5
do litel_8
release all except *p
clear
deactivate window cover
return

```

```

procedure litel_1
m_lamp=lamp
m_watts=watts
m_nwatts=nwatts
m_cost=cost
m_ncost=ncost
m_life=life
m_nlife=nlife
m_lumens=lumens
m_nlumen=nlumens
close databases
deactivate popup
return

```

```

procedure litel_2
activate window cover
@ 1,7 SAY "ECO: REPLACE STANDARD FLUORESCENT " +;
      "LAMPS WITH ENERGY EFFICIENT LAMPS"
@ 2,1 TO 7,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
@ 3,3 SAY "Existing lamp type:" get m_lamp
clear gets

```


return

procedure litel_3

```
@ 4,3 SAY "Audit code:" GET a code
@ 4,33 SAY "File name: L1" GET f_name pict "XXXXXX"
@ 4,53 say ".MEM"
@ 4,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 5,3 SAY "Electricity consumption cost:"
@ 5,33 SAY "$" get e_costp
@ 5,42 SAY "/kWh"
@ 5,48 SAY "Demand cost:"
@ 5,61 SAY "$" get d_costp
@ 5,68 SAY "/kW-month"
@ 6,3 SAY "Number of lamps:" get n_lamp picture "99999"
@ 6,38 SAY "Operating hours per year:" get o_hours PICT
"99999"
read
return
```

procedure litel_3a

move window cover2 to 5,5

activate window cover2

```
@ 1,3 say "
@ 2,3 say "----- LAMP DATA"
-----"
@ 3,3 SAY "LAMP WATTAGE:"
@ 3,18 SAY "Present:" get m_watts pict "9999"
@ 3,32 SAY "W"
@ 3,43 SAY "Proposed:" get m_nwatts pict "9999"
@ 3,58 SAY "W"
@ 5,3 SAY "LAMP COST:"
@ 5,18 SAY "Present:"
@ 5,27 SAY "$" get m_cost pict "999.99"
@ 5,43 SAY "Proposed:"
@ 5,53 SAY "$" get m_ncost pict "999.99"
@ 7,3 SAY "LAMP LIFE:"
@ 7,18 SAY "Present:" get m_life pict "99999"
@ 7,33 SAY "hrs"
@ 7,43 SAY "Proposed:" get m_nlife pict "99999"
@ 7,59 SAY "hrs"
@ 9,3 SAY "LAMP OUTPUT:"
@ 9,18 SAY "Present:" get m_lumens pict "99999"
@ 9,33 SAY "lumens"
@ 9,43 SAY "Proposed:" get m_nlumen pict "99999"
@ 9,59 SAY "lumens"
@ 10,3 say "-----" read
-----" read
return
```

procedure litel_4

do while .t.

do litel_3

an= " "

set colo to rg*/b+

```

@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
  return
endif
if upper(an)<>"N"
  loop
endif
exit
enddo
return

```

```

procedure lite1_4a
do while .t.
do lite1_3a
an= " "
set colo to rg*/b+
@ 17,23 say " ANY CHANGES? (Y/N) " get an pict "!"
read
set colo to w+/b+
@ 17,23 say space(30)
set colo to w+/b+
if lastkey()=27
  deactivate window cover2
  return
endif
if upper(an)<>"N"
  loop
endif
deactivate window cover2
exit
enddo
return

```

```

procedure lite1_5
dm_svngs=(m_watts-m_nwatts)/1000*val(n_lamp)
kw_svngs=round(dm_svngs*val(o_hours),0)
mb_svngs=kw_svngs*3412/1000000
d_svngs=round((kw_svngs*val(e_costp))+(dm_svngs*12*val(d_cost
p)),0) im_cost=round((m_ncost-m_cost)*val(n_lamp),0)
p_back=im_cost/d_svngs

```

```

@ 8,1 TO 19,78 DOUBLE
@ 8,33 SAY " CALCULATIONS "
@ 9,3 SAY "Lamp wattage"
@ 9,16 SAY "->"
@ 9,23 SAY "Present:" get m_watts pict "9999"
@ 9,37 SAY "W"
@ 9,50 SAY "Proposed:" get m_nwatts pict "9999"
@ 9,65 SAY "W"
@ 10,3 SAY "Lamp cost"

```

```

@ 10,16 SAY "->"
@ 10,23 SAY "Present:"
@ 10,32 SAY "$" get m_cost pict "999.99"
@ 10,50 SAY "Proposed:"
@ 10,60 SAY "$" get m_ncost pict "999.99"
@ 11,3 SAY "Lamp life"
@ 11,16 SAY "->"
@ 11,23 SAY "Present:" get m_life pict "99999"
@ 11,38 SAY "hrs"
@ 11,50 SAY "Proposed:" get m_nlife pict "99999"
@ 11,66 SAY "hrs"
@ 12,3 SAY "Lamp output"
@ 12,16 SAY "->"
@ 12,23 SAY "Present:" get m_lumens pict "99999"
@ 12,38 SAY "lumens"
@ 12,50 SAY "Proposed:" get m_nlumen pict "99999"
@ 12,66 SAY "lumens"
@ 14,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 14,26 SAY "kWh/yr"
@ 14,36 get mb_svngs picture "9999.99"
@ 14,44 SAY "MMBtu/yr"
@ 15,3 SAY "Dollar savings:"
@ 15,19 SAY "$" get d_svngs picture "999999"
@ 15,28 SAY "/yr"
@ 16,3 SAY "Implementation cost: $" get im_cost picture
"999999"
@ 16,35 SAY "(Incremental-replace as and when they fail)"
@ 17,3 SAY "Simple payback:" get p_back picture "99.9"
@ 17,24 SAY "years"
clear gets
return

```

```

procedure litel_6
an= " "
set colo to rg*/b+
@ 23,28 say "CHANGE LAMP TYPE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an)<>"N"
use litel
activate popup light1
if lastkey()<>13
return
endif
endif
return

```

```

procedure litel_7

```

```

move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: L1" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("l1"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "L1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
restore from "l1"+trim(mem_file) additive
do litel_2
do litel_3
do litel_6
do litel_2
do litel_4
do litel_4a
do litel_5
do litel_8
release all except *p
clear
deactivate window cover
return

procedure litel_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
  return
endif
if upper(an)<>"N"
  if file ("l1"+trim(f_name)+".mem")=.t.
    move window cover1 to 18,5
    activate window cover1
    @ 1,20 say "FILE L1"+upper(trim(f_name))+".MEM ALREADY
    EXISTS"
    @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
    "!"
    read

```

```

deactivate window cover1
  if upper (an)<>"N"
  delete file "L1"+trim(f_name)+".MEM"
  save to "l1"+trim(f_name)
  endif
else
  save to "l1"+trim(f_name)
endif
endif
return

procedure LITE1_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: L1" get
mem_file pict "XXXXXX" @ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("L1"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "L1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!".
  read
  deactivate window cover1
  return
endif
restore from "L1"+trim(mem_file) additive
define popup fp_LITE1 from 15,15 to 24,63;
shadow
define bar 4 of fp_LITE1 prompt "                PRINT TO
PRINTER"
define bar 6 of fp_LITE1 prompt "                PRINT TO DOS FILE
filename.txt"
on selection popup fp_LITE1 do p_LITE1
activate popup fp_LITE1
return

procedure p_LITE1
do case
case bar()=4
  move window cover1 to 17,5
  activate window cover1
  @ 2,20 say "PLEASE WAIT.....PRINTING "
  set print on
  set console off
  do pg_LITE1
  eject
  set print off
  set console on

```

```

release all except *p
deactivate window cover1
return
case bar()=6
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print to "L1"+trim(f_name)+".txt"
set print on
set console off
do pg_LITE1
set print off
set print to
set console on
release all except *p
deactivate window cover1
return
endcase
return

```

```

procedure pg_LITE1

```

```

?
?
?
?
? "      Audit code: "+trim(a_code)+"      Date: "+datp
? "      File name: L1"+upper(trim(f_name))
?
? "      ECO: REPLACE STANDARD FLUORESCENT LAMPS
WITH"
? "      ENERGY EFFICIENT FLUORESCENT LAMPS IN EXISTING
FIXTURES"
?
? "      RECOMMENDED ACTION"
?
? "      Energy-efficient (EE) fluorescent lamps consume
less energy"
? "      than standard lamps while giving nearly the same
light"
? "      levels. They may cost a little more, but the
additional"
? "      expense is recovered through energy savings.
These energy-"
? "      efficient lamps do not require fixture
modification or"
? "      ballast upgrade. We recommend that you replace
the standard"
? "      fluorescent lamps with energy-efficient lamps as
the standard"
? "      lamps fail. The savings calculated above will
not be"
? "      realized until all existing lamps have failed."
?

```

```

? "          DATA"
?
? "          Existing lamp type:           "+m_lamp
? "          Number of lamps:              "+(n_lamp)
? "          Operating hours per year:      "+(o_hoursp)+"
hours/yr"
? "          Electricity consumption cost: $" +(e_costp)+"
/kWh"
? "          Electricity demand cost:      $" +(d_costp)+" /kW-
month"
?
? "          Existing lamp data:"
? "          Wattage:      "+str(m_watts,8)+" W"
? "          Cost:        $" +str(m_cost,8,2)+" /lamp"
? "          Life:        "+str(m_life,8)+" hours"
? "          Output:      "+str(m_lumens,8)+" lumens"
?
? "          Proposed lamp data:"
? "          Wattage:      "+str(m_nwatts,8)+" W"
? "          Cost:        $" +str(m_ncost,8,2)+" /lamp"
? "          Life:        "+str(m_nlife,8)+" hours"
? "          Output:      "+str(m_nlumen,8)+" lumens"
?
? "          CALCULATIONS"
?
? "          Energy savings      =      "+str(kw_svngs,8)+ "
kWh/yr"
?
? "          =      "+str(mb_svngs,8,2)+ "
MMBtu/yr"
?
? "          Dollar savings      = $ "+str(d_svngs,8)+ " /yr"
?
? "          Implementation cost = $ "+str(im_cost,8)
?
? "          Simple payback      =      "+str(p_back,8,2)+"
years"
? " "
return

```

**** PROGRAM lite2.PRG

```
mem_file=space(6)
m_watts=192
m_nwatts=70
m_ncost=9.88
m_bcost=32.50
m_rcost=27
m_lcost=15
a_code=space(16)
f_name=space(6)
n_fix=space(5)
dm_svngs=space(6)
kw_svngs=space(6)
mb_svngs=space(6)
d_svngs=space(6)
im_cost=space(6)
p_back=space(3)
```

```
define window cover1 from 1,1 to 6,70 double;
  shadow
define window cover2 from 1,1 to 23,68 double;
  shadow
```

```
define popup fl_lite2 from 15,15 to 24,63;
shadow
define bar 2 of fl_lite2 prompt "          OPEN A NEW
FILE"
define bar 4 of fl_lite2 prompt "          DISPLAY NAMES OF
EXISTING FILES"
define bar 6 of fl_lite2 prompt "          RETRIEVE AN
EXISTING FILE"
define bar 8 of fl_lite2 prompt "          PRINT A
FILE"
on selection popup fl_lite2 do l_lite2

define popup fm_lite2 from 8,30 to 17,55;
prompt files like l2*.mem;
shadow
on selection popup fm_lite2 do m_lite2
```

```
activate popup fl_lite2
return
```

```
procedure l_lite2
do case
  case bar()=2
    do lite2 nw
  case bar()=4
    activate popup fm_lite2
  case bar()=6
```



```

do lite2_7
case bar()=8
do lite2_9

endcase
return

procedure m_lite2
deactivate popup
return

procedure lite2_nw
do lite2_2
do lite2_4
do lite2_4a
do lite2_5
do lite2_8
release all except *p
clear
deactivate window cover
return

procedure lite2_2
activate window cover
@ 1,7 SAY "ECO: REPLACE STANDARD FLUORESCENT F40T12 4 LAMP
FIXTURES WITH 2    "
@ 2,7 SAY "        HIGH EFFICIENCY, HIGH INTENSITY, LONG LIFE,
LAMPS, AN ELECTRONIC"
@ 3,7 SAY "        BALLAST, AND A HIGH EFFICIENCY REFLECTOR "
@ 4,1 TO 8,78 DOUBLE
@ 4,34 SAY " AUDIT DATA "
clear gets
return

procedure lite2_3
@ 5,3 SAY "Audit code:" GET a code
@ 5,33 SAY "File name: L2" GET f_name pict "XXXXXX"
@ 5,53 say ".MEM"
@ 5,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 6,3 SAY "Electricity consumption cost:"
@ 6,33 SAY "$" get e_costp
@ 6,42 SAY "/kWh"
@ 6,48 SAY "Demand cost:"
@ 6,61 SAY "$" get d_costp
@ 6,68 SAY "/kW-month"
@ 7,3 SAY "Number of fixtures:" get n_fix picture "99999"
@ 7,38 SAY "Operating hours per year:" get o_hoursp PICT
"9999"
read
return

procedure lite2_3a

```

```

move window cover2 to 2,5
activate window cover2
@ 1,3 say "
@ 2,3 say "-----"
-----"
@ 3,3 SAY "PRESENT LIGHTING SYSTEM: Fluorescent, 4F40T12 (48
inch)"
@ 4,3 SAY "Number of lamps per fixture ..... 4"
@ 5,3 SAY "Watts per fixture ..... " get
m_watts pict "9999"
@ 5,53 SAY "W "
@ 6,3 SAY "Lumens per fixture ..... 3,050
lumens "
@ 7,3 SAY "Lamp life .....
20,000 hours"
@ 8,3 SAY "Lamp cost ..... $
1.77 /lamp"
@ 9,3 say "-----"
-----"
@ 10,3 SAY "PROPOSED LIGHTING SYSTEM: Fluorescent, 4F40AX35
(48 inch)"
@ 11,3 SAY "Number of lamps per fixture ..... 2"
@ 12,3 SAY "Watts per fixture ..... " get
m_nwatts pict "9999"
@ 12,53 SAY "W "
@ 13,3 SAY "Lumens per fixture .....
3,700 lumens "
@ 14,3 SAY "Lamp life .....
24,000 hours"
@ 15,3 SAY "Lamp cost ..... $"
get m_ncost pict "999.99" @ 15,57 SAY "/lamp"
@ 16,3 SAY "Reflector cost per fixture ..... $"
get m_rcost pict "999.99" @ 17,3 SAY "Electronic ballast cost
per fixture ..... $" get m_bcost pict "999.99" @ 18,3 SAY
"Labor cost for retrofitting per fixture .... $" get m_lcost
pict "999.99" @ 19,3 say "-----"
-----"
read
return

```

```

procedure lite2_4
do while .t.
do lite2_3
an= " "
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an)<>"N"

```

```

loop
endif
exit
enddo
return

```

```

procedure lite2_4a
do while .t.
do lite2_3a
an= " "
set colo to rg*/b+
@ 20,23 say " ANY CHANGES? (Y/N) " get an pict "!"
read
set colo to w+/b+
@ 20,23 say space(30)
set colo to w+/b+
if lastkey()=27
deactivate window cover2
return
endif
if upper(an)<>"N"
loop
endif
deactivate window cover2
exit
enddo
return

```

```

procedure lite2_5
dm_svngs=(m_watts-m_nwatts)/1000*val(n_fix)
kw_svngs=round(dm_svngs*val(o_hoursp),0)
mb_svngs=kw_svngs*3412/1000000
d_svngs=round((kw_svngs*val(e_costp))+(dm_svngs*12*val(d_costp)),0)
im_cost=round((m_ncost*2+m_bcost+m_rcost+m_lcost)*val(n_fix),0)
p_back=round(im_cost/d_svngs,1)

```

```

@ 9,1 TO 16,78 DOUBLE
@ 9,33 SAY " CALCULATIONS "
@ 11,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 11,26 SAY "kWh/yr"
@ 11,36 get mb_svngs picture "9999.99"
@ 11,44 SAY "MMBtu/yr"
@ 12,3 SAY "Dollar savings:"
@ 12,19 SAY "$" get d_svngs picture "999999"
@ 12,28 SAY "/yr"
@ 13,3 SAY "Implementation cost: $" get im_cost picture
"999999"
@ 14,3 SAY "Simple payback:" get p_back picture "99.9"
@ 14,24 SAY "years"
clear gets
return

```

```

procedure lite2_7

```

```

move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: L2" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("l2"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "L2"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
restore from "l2"+trim(mem_file) additive
do lite2_2
do lite2_3
do lite2_4
do lite2_4a
do lite2_5
do lite2_8
release all except *p
clear
deactivate window cover
return

procedure lite2_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
  return
endif
if upper(an)<>"N"
  if file ("l2"+trim(f_name)+".mem")=.t.
    move window cover1 to 18,5
    activate window cover1
    @ 1,20 say "FILE L2"+upper(trim(f_name))+".MEM ALREADY
    EXISTS"
    @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
    "!"
    read
    deactivate window cover1
    if upper (an)<>"N"

```

```

    delete file "L2"+trim(f_name)+".MEM"
    save to "l2"+trim(f_name)
endif
else
    save to "l2"+trim(f_name)
endif
endif
return

procedure LITE2_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: L2" get
mem file pict "XXXXXX"
@ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
    return
endif
if file ("L2"+trim(mem_file)+".mem")<>.t.
    move window cover1 to 17,5
    activate window cover1
    an=" "
    @ 1,20 say "L2"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
    @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
    read
    deactivate window cover1
    return
endif
restore from "L2"+trim(mem_file) additive
define popup fp_LITE2 from 15,15 to 24,63;
shadow
define bar 4 of fp_LITE2 prompt "                PRINT TO
PRINTER"
define bar 6 of fp_LITE2 prompt "                PRINT TO DOS FILE
filename.txt"
on selection popup fp_LITE2 do p_LITE2
activate popup fp_LITE2
return

procedure p_LITE2
do case
case bar()=4
    move window cover1 to 17,5
    activate window cover1
    @ 2,20 say "PLEASE WAIT.....PRINTING "
    set print on
    set console off
    do pg_LITE2
    eject
    set print off
    set console on
    release all except *p

```

```

deactivate window cover1
return
case bar()=6
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print to "L2"+trim(f_name)+".txt"
set print on
set console off
do pg_LITE2
set print off
set print to
set console on
release all except *p
deactivate window cover1
return
endcase
return

```

```

procedure pg_LITE2

```

```

?
?
?
?
?
?
? "      Audit code: "+trim(a_code)+"      Date: "+datp
? "      File name: L2"+upper(trim(f_name))
?
? "      ECO: REPLACE STANDARD FLUORESCENT F40T12 4 LAMP
FIXTURES WITH"
? "      2 HIGH EFFICIENCY, HIGH INTENSITY, LONG
LIFE, LAMPS,"
? "      AN ELECTRONIC BALLAST, AND A HIGH
EFFICIENCY REFLECTOR "
?
? "      RECOMMENDED ACTION"
?
? "      Energy savings can be achieved by retrofitting
existing "
? "      fluorescent lighting fixtures. The four lamp
(48 inch)"
? "      fixtures currently in place can be replaced by
high      "
? "      efficiency, high intensity, long life two lamp
fixtures  "
? "      which will save energy while retaining the same
light level."
? "      This is accomplished by adding a reflector along
with the  "
? "      lamps in order to point the maximum amount of
light in the "
? "      direction needed. Each retrofit fixture will
consist of an"
? "      electronic ballast, a reflector, and two lamp

```

```

receptacles."
? "          DATA"
?
? "          Number of fixtures:          "+(n_fix)
? "          Operating hours per year:     "+(o_hoursp)+"
hours/yr"
? "          Electricity consumption cost:$"+(e_costp)+"
/kWh"
? "          Electricity demand cost:      $"+(d_costp)+" /kW-
month"
?
? "          Present lighting system: Fluorescent, 4F40T12
(48 inch)"
? "          Number of lamps per fixture .... 4"
? "          Watts per fixture
....."+str(m_watts,8)+" W"
? "          Lumens per fixture ..... 3,050 lumens "
? "          Lamp life ..... 20,000 hours"
? "          Lamp cost ..... $ 1.77 /lamp"
?
? "          Proposed lighting system: Fluorescent, 4F40AX35
(48 inch)"
? "          Number of lamps per fixture ..... 2"
? "          Watts per fixture
....."+str(m_nwatts,8)+" W"
? "          Lumens per fixture .....
3,700 lumens "
? "          Lamp life .....
24,000 hours"
? "          Lamp cost .....
$"+str(m_ncost,8,2)+" /lamp"
? "          Reflector cost per fixture .....
$"+str(m_rcost,8,2)
? "          Electronic ballast cost per fixture .....
$"+str(m_bcost,8,2)
? "          Labor cost for retrofitting per fixture ....
$"+str(m_lcost,8,2)
?
? "          CALCULATIONS"
?
? "          Energy savings          =  "+str(kw_svngs,8)+ "
kWh/yr"
?
? "          =  "+str(mb_svngs,8,2)+ "
MMBtu/yr"
?
? "          Dollar savings          = $ "+str(d_svngs,8)+ " /yr"
?
? "          Implementation cost = $ "+str(im_cost,8)
?
? "          Simple payback          =  "+str(p_back,8,2)+"
years"
? " "
return

```

**** PROGRAM compl.prg ****

```
mem_file=space(6)
m_pres=space(6)
m_vl_kwh=space(6)
m_l_kwh=space(6)
m_m_kwh=space(6)
m_s_kwh=space(6)
a_code=space(16)
f_name=space(6)
vl_leak=space(3)
l_leak=space(3)
m_leak=space(3)
s_leak=space(3)
nr_leak=space(3)
p_cent=90
kw_svngs=space(6)
mb_svngs=space(6)
d_svngs=space(6)
im_cost=space(6)
nd_svngs=space(6)
```

```
define window cover1 from 1,1 to 6,70 double;
  shadow
```

```
define popup fl_comp1 from 15,15 to 24,63;
  shadow
```

```
define bar 2 of fl_comp1 prompt "          OPEN A NEW
FILE"
define bar 4 of fl_comp1 prompt "          DISPLAY NAMES OF
EXISTING FILES"
define bar 6 of fl_comp1 prompt "          RETRIEVE AN
EXISTING FILE"
define bar 8 of fl_comp1 prompt "          PRINT A
FILE"
```

```
on selection popup fl_comp1 do l_comp1
```

```
define popup fm_comp1 from 8,30 to 17,55;
  prompt files like c1*.mem;
  shadow
  on selection popup fm_comp1 do m_comp1
```

```
activate popup fl_comp1
return
```

```
procedure l_comp1
do case
  case bar()=2
    do compl_nw
  case bar()=4
    activate popup fm_comp1
```



```

case bar()=6
  do compl_7
case bar()=8
  do compl_9
endcase
return

```

```

procedure m_compl
deactivate popup
return

```

```

procedure compl_nw
use compl_d
activate popup compl_p
if lastkey() <> 13
  return
endif
do compl_2
do compl_4
do compl_5
do compl_8
release all except *p
clear
deactivate window cover
return

```

```

procedure compl_od
activate popup fn_compl
return

```

```

procedure compl_1
m_pres=pres
m_vl_kwh=vl_kwh
m_l_kwh=l_kwh
m_m_kwh=m_kwh
m_s_kwh=s_kwh
close databases
deactivate popup
return

```

```

procedure compl_2
activate window cover
@ 1,7 SAY "          ECO: ELIMINATE COMPRESSED "+;
          "AIR LEAKS"
@ 2,1 TO 12,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
@ 3,3 SAY "Compressor discharge pressure:" get m_pres pict
"999"
clear gets
return

```

```

procedure compl_3
@ 4,3 SAY "Audit code:" GET a_code

```

```

@ 4,33 SAY "File name: C1" GET f_name pict "XXXXXX"
@ 4,53 say ".MEM"
@ 4,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 5,3 SAY "Electricity consumption cost:"
@ 5,33 SAY "$" get e_costp
@ 5,42 SAY "/kWh"
@ 6,3 SAY "Operating hours per year:" get o_hoursP pict
"9999"
@ 7,3 SAY "Number of very large leaks (1/4 inch):" get
vl_leak pict "99"
@ 8,3 SAY "Number of large leaks (1/8 inch):" get l_leak pict
"99"
@ 9,3 SAY "Number of medium leaks (1/16 inch):" get m_leak
pict "99"
@ 10,3 SAY "Number of small leaks (1/32 inch):" get s_leak
pict "99"
@ 11,3 SAY "Estimated percentage of air leak reduction from
repair:" get p cent pict "99"
@ 11,61 say "%"
read
return

```

```

procedure compl_4
do while .t.
do compl_3
an= " "
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an) <> "N"
loop
endif
exit
enddo
return

```

```

procedure compl_5
kw_svngs=round((val(vl_leak)*m_vl_kwh+val(l_leak)*m_l_kwh+val
(m_leak)*m_m_kwh+val(s_leak)*m_s_kwh)*val(o_hoursP)/78760*p_ce
nt/100,0) mb_svngs=kw_svngs*34127/1000000
d_svngs=round(kw_svngs*val(e_costp),0)
nr_leak
=round((val(vl_leak)+val(l_leak)+val(m_leak)+val(s_leak))/2,0
)
im_cost=nr_leak*10
nd_svngs=d_svngs-im_cost
@ 14,1 TO 22,78 DOUBLE

```

```

@ 14,33 SAY " CALCULATIONS "
@ 15,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 15,26 SAY "kWh/yr"
@ 15,36 get mb_svngs picture "9999.99"
@ 15,44 SAY "MMBtu/yr"
@ 16,3 SAY "Dollar savings:"
@ 16,19 SAY "$" get d_svngs picture "999999"
@ 16,28 SAY "/yr"
@ 17,3 SAY "Maintenance cost: $" get im_cost picture "999999"
@ 17,30 SAY "/yr"
@ 18,5 SAY "(Assuming repair of leaks per year at a cost of
$10/repair, after "
@ 18,24 get nr_leak picture "99"
@ 19,5 say "an initial effort to repair existing leaks)"
@ 20,3 say "Net dollar savings: $" get nd_svngs picture
"999999"
@ 20,32 SAY "/yr"
@ 21,3 SAY "Simple payback: Immediate"
clear gets
return

```

```

procedure compl_6
an= " "
set colo to rg*/b+
@ 23,25 say "CHANGE COMPRESSOR PRESSURE? (Y/N)" get an pict
"! "
read
set colo to w+/b+
@ 23,25 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an) <> "N"
use compl_d
activate popup compl_p
if lastkey() <> 13
return
endif
endif
return

```

```

procedure compl_7
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: C1" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
return
endif

```

```

if file ("c1"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "C1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
restore from "c1"+trim(mem_file) additive
do compl_2
do compl_3
do compl_6
do compl_2
do compl_4
do compl_5
do compl_8
release all except *p
clear
deactivate window cover
return

procedure compl_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
  return
endif
if upper(an)<>"N"
  if file ("c1"+trim(f_name)+".mem")=.t.
    move window cover1 to 18,5
    activate window cover1
    @ 1,20 say "FILE C1"+upper(trim(f_name))+".MEM ALREADY
    EXISTS"
    @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
    "!"
    read
    deactivate window cover1
    if upper (an)<>"N"
      delete file "C1"+trim(f_name)+".MEM"
      save to "c1"+trim(f_name)
    endif
  else
    save to "c1"+trim(f_name)
  endif
endif
return

```

```

procedure COMP1_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: C1" get
mem_file pict "XXXXXX"
@ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
return
endif
if file ("C1"+trim(mem_file)+".mem")<>.t.
move window cover1 to 17,5
activate window cover1
an=" "
@ 1,20 say "C1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
@ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
read
deactivate window cover1
return
endif
restore from "C1"+trim(mem_file) additive
define popup fp_COMP1 from 15,15 to 24,63;
shadow
define bar 4 of fp_COMP1 prompt " PRINT TO
PRINTER"
define bar 6 of fp_COMP1 prompt " PRINT TO DOS FILE
filename.txt"
on selection popup fp_COMP1 do p_COMP1
activate popup fp_COMP1
return

procedure p_COMP1
do case
case bar()=4
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print on
set console off
do pg_COMP1
eject
set print off
set console on
release all except *p
deactivate window cover1
return
case bar()=6
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print to "C1"+trim(f_name)+".txt"
set print on
set console off

```

```

do pg_COMP1
set print off
set print to
set console on
release all except *p
deactivate window cover1
return
endcase
return

```

```

procedure pg_COMP1

```

```

?
?
?
?
? "          Audit code: "+trim(a_code)+"      Date: "+datp
? "          File name: C1"+upper(trim(f_name))
?
? "          ECO: ELIMINATE COMPRESSED AIR
LEAKS"
?
? "          RECOMMENDED ACTION"
?
? "          Compressed air is distributed through pipelines
from the "          " compressors to the pneumatic equipment. Because
? "          of the "          connections in the piping and accidental holes
? "          in the tubing,"          air leaks are unavoidable. However, an ongoing
? "          maintenance"          program to repair leaks can reduce their number
? "          and make them"          short-lived. By fixing the air leaks energy
? "          savings can be "          achieved. "
?
? "          DATA"
?
? "          Compressor discharge pressure:"+str(m_pres,4)+"
psig"
? "          Operating hours per year:      "+(o_hoursp)+"
hours/yr"
? "          Electricity consumption cost:$"+(e_costp)+"
/kWh"
? "          Estimated air leak reduction from
repair:"+str(p_cent,3)+" %"
? "          Number of very large leaks (1/4
inch):"+(vl_leak)
? "          Number of large leaks (1/8 inch):      "+(l_leak)
? "          Number of medium leaks (1/16 inch):    "+(m_leak)
? "          Number of small leaks (1/32 inch):     "+(s_leak)
?

```

```

? "          CALCULATIONS"
?
? "          Energy savings      = "+str(kw_svngs,8)+ "
kWh/yr"
? "          = "+str(mb_svngs,8,2)+ "
MMBtu/yr"
?
? "          Dollar savings      = $ "+str(d_svngs,8)+ " /yr"
?
? "          Maintenance cost    = $ "+str(im_cost,8)+ " /yr"
? "          (Assuming repair of "+str(nr_leak,4)+" leaks
per year"
? "          at a cost of $10/repair, after an initial
effort to"
? "          repair existing leaks)"
?
? "          Net dollar savings  = $ "+str(nd_svngs,8)+" /yr"
?
? "          Simple payback      = Immediate"
? " "
return

```

**** PROGRAM comp2.PRG

```
mem_file=space(6)
a_code=space(16)
f_name=space(6)
hp=0
hpd=0
load=90
pres=0
dm_svngs=0
kw_svngs=0
mb_svngs=0
d_svngs=0
im_cost="NIL"
p_back="IMMEDIATE"
```

```
hpd100=0
hpd110=5
hpd120=10
hpd130=16.5
hpd140=23
hpd150=30
```

```
pl=0
```

```
define window cover1 from 1,1 to 6,70 double;
  shadow
```

```
define popup fl_comp2 from 15,15 to 24,63;
shadow
```

```
define bar 2 of fl_comp2 prompt "                OPEN A NEW
FILE"
```

```
define bar 4 of fl_comp2 prompt "                DISPLAY NAMES OF
EXISTING FILES"
```

```
define bar 6 of fl_comp2 prompt "                RETRIEVE AN
EXISTING FILE"
```

```
define bar 8 of fl_comp2 prompt "                PRINT A
FILE"
```

```
on selection popup fl_comp2 do l_comp2
```

```
define popup fm_comp2 from 8,30 to 17,55;
```

```
prompt files like c2*.mem;
```

```
shadow
```

```
on selection popup fm_comp2 do m_comp2
```

```
activate popup fl_comp2
return
```

```
procedure l_comp2
```

```
do case
```

```
  case bar()=2
```



```

do comp2_nw
case bar()=4
activate popup fm_comp2
case bar()=6
do comp2_7
case bar()=8
do comp2_9

endcase
return

procedure m_comp2
deactivate popup
return

procedure comp2_nw
do comp2_2
do comp2_4
do comp2_5
do comp2_8
release all except *p
clear
deactivate window cover
return

procedure comp2_2
activate window cover
@ 1,7 SAY " ECO: REDUCE COMPRESSED AIR PRESSURE TO
MINIMUM REQUIRED"

@ 2,1 TO 9,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
clear gets
return

procedure comp2_3
@ 3,3 SAY "Audit code:" GET a code
@ 3,33 SAY "File name: C2" GET f_name pict "XXXXXX"
@ 3,53 say ".MEM"
@ 3,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 4,3 SAY "Electricity consumption cost:"
@ 4,33 SAY "$" get e_costp
@ 4,42 SAY "/kWh"
@ 4,48 SAY "Demand cost:"
@ 4,61 SAY "$" get d_costp
@ 4,68 SAY "/kW-month"
@ 5,3 SAY "Operating hours per year:" get o_hoursP pict
"9999"
@ 5,34 say "hrs/yr"
@ 6,3 say "Total air compressor HP:" get hp pict "999"
@ 6,32 say "HP"

```

```

@ 7,3 say "Air compressor load factor:" get load pict "99"
@ 7,34 say "%"
@ 8,3 say "Present compressed air pressure:" get pres pict
"999" range 100,150 @ 8,40 say "psig"
read
return

```

```

procedure comp2_4
do while .t.
do comp2_3
an= " "
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an) <> "N"
loop
endif
exit
enddo
return

```

```

procedure comp2_5
hpdl=0
hpdu=0

if pres >=100 .and. pres <110
pl=100
hpdl=0
hpdu=5
endif
if pres >=110 .and. pres <120
pl=110
hpdl=5
hpdu=10
endif
if pres >=120 .and. pres <130
pl=120
hpdl=10
hpdu=16.5
endif
if pres >=130 .and. pres <140
pl=130
hpdl=16.5
hpdu=23
endif
if pres >=140 .and. pres <=150

```

```

pl=140
hpd1=23
hpdu=30
endif

hpd=0
hpd=round(hpd1+(hpdu-hpd1)/10*(pres-pl),1)

dm_svngs=hp*load/100*hpd/100*.746
kw_svngs=round(dm_svngs*val(o_hours_p),0)
mb_svngs=kw_svngs*3412/1000000
d_svngs=round((kw_svngs*val(e_cost_p))+(dm_svngs*12*val(d_cost_p)),0)

@ 10,1 TO 18,78 DOUBLE
@ 10,33 SAY " CALCULATIONS "
@ 11,3 say "Approximate percentage decrease in BHP due to
reduction "
@ 12,3 say " in pressure of compressed air to 100 psig :"
get hpd pict "99.9" @ 12,54 say "%"
@ 14,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 14,26 SAY "kWh/yr"
@ 14,36 get mb_svngs picture "9999.99"
@ 14,44 SAY "MMBtu/yr"
@ 15,3 SAY "Dollar savings:"
@ 15,19 SAY "$" get d_svngs picture "999999"
@ 15,28 SAY "/yr"
@ 16,3 SAY "Implementation cost: " get im_cost
@ 17,3 SAY "Simple payback:" get p_back
clear gets
return

procedure comp2_7
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: C2" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
return
endif
if file ("c2"+trim(mem_file)+".mem")<>.t.
move window cover1 to 17,5
activate window cover1
an=" "
@ 1,20 say "C2"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
@ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
read
deactivate window cover1
return
endif
endif

```

```

restore from "c2"+trim(mem_file) additive
do comp2_2
do comp2_3
do comp2_4
do comp2_5
do comp2_8
release all except *p
clear
deactivate window cover
return

```

```

procedure comp2 8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an)<>"N"
if file ("c2"+trim(f_name)+".mem")=.t.
move window cover1 to 18,5
activate window cover1
@ 1,20 say "FILE C2"+upper(trim(f_name))+".MEM ALREADY
EXISTS"
@ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
"! "
read
deactivate window cover1
if upper (an)<>"N"
delete file "C2"+trim(f_name)+".MEM"
save to "c2"+trim(f_name)
endif
else
save to "c2"+trim(f_name)
endif
endif
return

```

```

procedure COMP2 9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: C2" get
mem_file pict "XXXXXX" @ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
return
endif
if file ("C2"+trim(mem_file)+".mem")<>.t.
move window cover1 to 17,5

```

```

activate window cover1
an=" "
@ 1,20 say "C2"+upper(trim(mem file))+".MEM DOES NOT EXIST"
@ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
read
deactivate window cover1
return
endif
restore from "C2"+trim(mem file) additive
define popup fp_COMP2 from 15,15 to 24,63;
shadow
define bar 4 of fp_COMP2 prompt " PRINT TO
PRINTER"
define bar 6 of fp_COMP2 prompt " PRINT TO DOS FILE
filename.txt"
on selection popup fp_COMP2 do p_COMP2
activate popup fp_COMP2
return

procedure p_COMP2
do case
case bar()=4
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print on
set console off
do pg_COMP2
eject
set print off
set console on
release all except *p
deactivate window cover1
return
case bar()=6
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print to "C2"+trim(f_name)+".txt"
set print on
set console off
do pg_COMP2
set print off
set print to
set console on
release all except *p
deactivate window cover1
return
endcase
return

procedure pg_COMP2
?
?
```

```

?
?
?
? "      Audit code: "+trim(a_code)+"      Date: "+datp
? "      File name: C2"+upper(trim(f_name))
?
? "      ECO: REDUCE COMPRESSED AIR PRESSURE TO MINIMUM
REQUIRED"
?
? "      RECOMMENDED ACTION"
?
? "      Compressed air is distributed through pipelines
from the "
? "      compressors to the pneumatic equipment. The
discharge "
? "      pressure of compressed air system could be
lowered without "
? "      causing operating problems. This reduction in
air pressure"
? "      to the minimum required level reduces the energy
consumption "
? "      of the electric motors driving the compressors.
When implem-"
? "      enting this ECO it should be verified that the
lower "
? "      discharge pressure is acceptable for all of the
equipment in"
? "      the plant ."
?
? "      DATA"
?
? "      Total air compressor HP:          "+str(hp,8)+"
HP"
? "      Air compressor load factor:       "+str(load,8)+"
%"
? "      Present compressed air pressure: "+str(pres,8)+"
psig"
? "      Operating hours per year:        "+(o_hoursp)+"
hours/yr"
? "      Electricity consumption cost:$"+(e_costp)+"
/kWh"
? "      Electricity demand cost:        $"+(d_costp)+" /kW-
month"
?
? "      CALCULATIONS"
?
? "      Approximate percentage decrease in BHP due to
reduction "
? "      in pressure of compressed air to 100 psig :
"+str(hpd,8,1)+" %"
?
? "      Energy savings          =      "+str(kw_svngs,8)+ "
kWh/yr"
?

```

```
? "
MMBtu/yr" = "+str(mb_svngs,8,2)+ "
?
? " Dollar savings = $ "+str(d_svngs,8)+ " /yr"
?
? " Implementation cost = Nil"
?
? " Simple payback = Immediate"
? " "
return
```

**** PROGRAM ELEC1.PRG

```
mem_file=space(6)
a_code=space(16)
f_name=space(6)
moteff=90
partload=80
kw_svngs=0
mb_svngs=space(6)
d_svngs=space(6)
im_cost=0
p_back=0
```

```
declare
desc[6],no_mot[6],capa[6],op_hour[6],v_imp[6],duty_rpm[8],duty_oph[8],est_eff[8]
n=1
do while n<7
desc[n]=space(20)
no_mot[n]=space(3)
capa[n]=space(3)
op_hour[n]=space(4)
v_imp[n]=0
n=n+1
enddo
```

```
duty_rpm[1]=100
duty_rpm[2]=90
duty_rpm[3]=80
duty_rpm[4]=70
duty_rpm[5]=60
duty_rpm[6]=50
duty_rpm[7]=40
duty_rpm[8]=30
```

```
duty_oph[1]=0
duty_oph[2]=3
duty_oph[3]=9
duty_oph[4]=21
duty_oph[5]=23
duty_oph[6]=23
duty_oph[7]=16
duty_oph[8]=5
```

```
est_eff[1]=82
est_eff[2]=81
est_eff[3]=80
est_eff[4]=76
est_eff[5]=70
est_eff[6]=66
est_eff[7]=64
est_eff[8]=54
```



```

define window cover1 from 1,1 to 6,70 double;
  shadow
define window cover2 from 1,1 to 20,68 double;
  shadow

define popup fl_elec1 from 15,15 to 24,63;
shadow
define bar 2 of fl_elec1 prompt "                OPEN A NEW
FILE"
define bar 4 of fl_elec1 prompt "                DISPLAY NAMES OF
EXISTING FILES"
define bar 6 of fl_elec1 prompt "                RETRIEVE AN
EXISTING FILE"
define bar 8 of fl_elec1 prompt "                PRINT A
FILE"
on selection popup fl_elec1 do l_elec1

define popup fm_elec1 from 8,30 to 17,55;
prompt files like e1*.mem;
shadow
on selection popup fm_elec1 do m_elec1

activate popup fl_elec1
return

procedure l_elec1
do case
  case bar()=2
    do elec1_nw
  case bar()=4
    activate popup fm_elec1
  case bar()=6
    do elec1_7
  case bar()=8
    do elec1_9

endcase
return

procedure m_elec1
deactivate popup
return

procedure elec1_nw
do elec1_2
do elec1_4
do elec1_4a
do elec1_4x
do elec1_4b

```

```

do elec1_5
do elec1_8
release all except *p
clear
deactivate window cover
return

```

```

procedure elec1_2
activate window cover
@ 1,7 SAY "      ECO: USE VARIABLE FREQUENCY DRIVES" +;
      " ON SELECTED MOTORS"
@ 2,1 TO 16,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
clear gets
return

```

```

procedure elec1_3
@ 3,3 SAY "Audit code:" GET a code
@ 3,33 SAY "File name: E1" GET f_name pict "XXXXXX"
@ 3,53 say ".MEM"
@ 3,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 4,3 SAY "Electricity consumption cost:"
@ 4,33 SAY "$" get e_costp
@ 4,42 SAY "/kWh"
@ 6,3 SAY "Estimated motor efficiency:" get moteff pict "99"
@ 6,33 say "%"
@ 6,38 SAY "Estimated motor partload factor:" get partload
pict "99"
@ 6,73 say "%"
@ 8,3 say "System      Description      Number of      Motor
Operating hours"
@ 9,3 say " #
capacity (HP)      hours/yr      "
@ 10,3 say " 1"
@ 10,12 get desc[1]
@ 10,37 get no_mot[1]
@ 10,51 get capa[1]
@ 10,67 get op_hour[1]
@ 11,3 say " 2"
@ 11,12 get desc[2]
@ 11,37 get no_mot[2]
@ 11,51 get capa[2]
@ 11,67 get op_hour[2]
@ 12,3 say " 3"
@ 12,12 get desc[3]
@ 12,37 get no_mot[3]
@ 12,51 get capa[3]
@ 12,67 get op_hour[3]
@ 13,3 say " 4"
@ 13,12 get desc[4]
@ 13,37 get no_mot[4]
@ 13,51 get capa[4]

```

```

@ 13,67 get op_hour[4]
@ 14,3 say " 5"
@ 14,12 get desc[5]
@ 14,37 get no_mot[5]
@ 14,51 get capa[5]
@ 14,67 get op_hour[5]
@ 15,3 say " 6"
@ 15,12 get desc[6]
@ 15,37 get no_mot[6]
@ 15,51 get capa[6]
@ 15,67 get op_hour[6]
read
return

```

```

procedure elec1_3a
move window cover2 to 5,5
activate window cover2

```

```

@ 1,3 say " DUTY CYCLE "
@ 2,3 say " -----"
-----"
@ 3,3 say " % Rated % Operating Estimated
system "
@ 4,3 say " RPM hours
efficiency"
@ 5,3 say " -----"
-----"
@ 6,10 get duty_rpm[1] pict "999"
@ 6,28 get duty_oph[1] pict "99"
@ 6,49 get est_eff[1] pict "99"
@ 7,10 get duty_rpm[2] pict "999"
@ 7,28 get duty_oph[2] pict "99"
@ 7,49 get est_eff[2] pict "99"
@ 8,10 get duty_rpm[3] pict "999"
@ 8,28 get duty_oph[3] pict "99"
@ 8,49 get est_eff[3] pict "99"
@ 9,10 get duty_rpm[4] pict "999"
@ 9,28 get duty_oph[4] pict "99"
@ 9,49 get est_eff[4] pict "99"
@ 10,10 get duty_rpm[5] pict "999"
@ 10,28 get duty_oph[5] pict "99"
@ 10,49 get est_eff[5] pict "99"
@ 11,10 get duty_rpm[6] pict "999"
@ 11,28 get duty_oph[6] pict "99"
@ 11,49 get est_eff[6] pict "99"
@ 12,10 get duty_rpm[7] pict "999"
@ 12,28 get duty_oph[7] pict "99"
@ 12,49 get est_eff[7] pict "99"
@ 13,10 get duty_rpm[8] pict "999"
@ 13,28 get duty_oph[8] pict "99"
@ 13,49 get est_eff[8] pict "99"
read
return

```

```

procedure elec1_3b

```

```

move window cover2 to 5,5
activate window cover2
@ 1,3 say "
@ 2,3 say "
IMPLEMENTATION COST "
-----"
@ 3,3 say " Description Motor VFD
implementation "
@ 4,3 say " capacity (HP) cost
($)"
@ 5,3 say "
-----"
@ 6,7 get desc[1]
@ 6,33 get capa[1]
@ 6,46 get v_imp[1]
@ 7,7 get desc[2]
@ 7,33 get capa[2]
@ 7,46 get v_imp[2]
@ 8,7 get desc[3]
@ 8,33 get capa[3]
@ 8,46 get v_imp[3]
@ 9,7 get desc[4]
@ 9,33 get capa[4]
@ 9,46 get v_imp[4]
@ 10,7 get desc[5]
@ 10,33 get capa[5]
@ 10,46 get v_imp[5]
@ 11,7 get desc[6]
@ 11,33 get capa[6]
@ 11,46 get v_imp[6]
@12,3 say "
-----"
@13,3 say " Note: Default VFD implementation cost is based
on a "
@14,3 say " reasonable estimate of $185 per motor HP,
based"
@15,3 say " on values obtained from vendor
consultation. "
read
return

```

```

procedure elec1_4
do while .t.
do elec1_3
an= " "
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif

```

```

if upper(an) <> "N"
  loop
endif
exit
enddo
return

procedure elec1_4a
do while .t.
do elec1_3a
an= " "
set colo to rg*/b+
@ 16,23 say " ANY CHANGES? (Y/N) " get an pict "!"
read
set colo to w+/b+
@ 16,23 say space(30)
set colo to w+/b+
if lastkey()=27
  deactivate window cover2
  return
endif
if upper(an) <> "N"
  loop
endif
deactivate window cover2
exit
enddo
return

procedure elec1_4x
v_cost=185
n=1
do while n<7
  v_imp[n]=round(v_cost*val(capa[n]),0)
  n=n+1
enddo
return

procedure elec1_4b
do while .t.
do elec1_3b
an= " "
set colo to rg*/b+
@ 17,23 say " ANY CHANGES? (Y/N) " get an pict "!"
read
set colo to w+/b+
@ 17,23 say space(30)
set colo to w+/b+
if lastkey()=27
  deactivate window cover2
  return
endif
if upper(an) <> "N"
  loop

```

```

endif
deactivate window cover2
exit
enddo
return

```

```

procedure elec1_5
n=1
pow1=0
kw_svngs=0
do while n<7
  pow1=100/moteff*val(op_hour[n])*partload/100*.746
  pow2=0
  k=1
  do while k<9
    pow2=pow2+duty_rpm[k]/100*duty_rpm[k]/100*duty_rpm[k]/100*
      .746/est_eff[k]*100*duty_oph[k]/100*val(op_hour[n]) k=k+1
  enddo
  kw_svngs=kw_svngs+round((pow1-
    pow2)*val(no_mot[n])*val(capa[n]),0)
  n=n+1
enddo
mb_svngs=kw_svngs*3412/1000000
d_svngs=round(kw_svngs*val(e_costp),0)
n=1
im_cost=0
do while n<7
  im_cost=im_cost+v_imp[n]*val(no_mot[n])
  n=n+1
enddo
p_back=round(im_cost/d_svngs,1)

@ 17,1 TO 22,78 DOUBLE
@ 17,33 SAY " CALCULATIONS "

@ 18,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 18,26 SAY "kWh/yr"
@ 18,36 get mb_svngs picture "9999.99"
@ 18,44 SAY "MMBtu/yr"
@ 19,3 SAY "Dollar savings:"
@ 19,19 SAY "$" get d_svngs picture "999999"
@ 19,28 SAY "/yr"
@ 20,3 SAY "Implementation cost: " get im_cost pict "999999"
@ 21,3 SAY "Simple payback:" get p_back pict "99.9"
clear gets
return

```

```

procedure elec1_7
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: E1" get mem_file

```

```

pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("e1"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "E1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
restore from "e1"+trim(mem_file) additive
do elec1_2
do elec1_3
do elec1_4
do elec1_4a
do elec1_4b
do elec1_5
do elec1_8
release all except *p
clear
deactivate window cover
return

procedure elec1_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
  return
endif
if upper(an)<>"N"
  if file ("e1"+trim(f_name)+".mem")=.t.
    move window cover1 to 18,5
    activate window cover1
    @ 1,20 say "FILE E1"+upper(trim(f_name))+".MEM ALREADY EXISTS"
    @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict "!"
  read
  deactivate window cover1
  if upper (an)<>"N"
    delete file "E1"+trim(f_name)+".MEM"
    save to "e1"+trim(f_name)
  
```

```

    endif
else
    save to "e1"+trim(f_name)
endif
endif
return

procedure elec1_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: E1" get
mem_file pict "XXXXXX"
@ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
    return
endif
if file ("e1"+trim(mem_file)+".mem")<>.t.
    move window cover1 to 17,5
    activate window cover1
    an=" "
    @ 1,20 say "E1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
    @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
    read
    deactivate window cover1
    return
endif
restore from "e1"+trim(mem_file)+additive
define popup fp_elec1 from 15,15 to 24,63;
shadow
define bar 4 of fp_elec1 prompt "                PRINT TO
PRINTER"
define bar 6 of fp_elec1 prompt "                PRINT TO DOS FILE
filename.txt"
on selection popup fp_elec1 do p_elec1
activate popup fp_elec1
return

procedure p_elec1
do case
case bar()=4
    move window cover1 to 17,5
    activate window cover1
    @ 2,20 say "PLEASE WAIT.....PRINTING "
    set print on
    set console off
    do pg_elec1
    eject
    set print off
    set console on
    release all except *p
    deactivate window cover1
    return

```



```
case bar()=6
  move window cover1 to 17,5
  activate window cover1
  @ 2,20 say "PLEASE WAIT.....PRINTING "
  set print to "e1"+trim(f_name)+".txt"
  set print on
  set console off
  do pg_elec1
  set print off
  set print to
  set console on
  release all except *p
  deactivate window cover1
  return
endcase
return
```

**** PROGRAM ELEC2.PRG

```
mem_file=space(6)
a_code=space(16)
f_name=space(6)
pCentsav=3
moteff=90
partload=80
dm_svngs=0
kw_svngs=0
mb_svngs=space(6)
d_svngs=space(6)
im_cost="NIL"
p_back="IMMEDIATE"
```

```
declare desc[6],no_mot[6],capa[6],op_hour[6]
n=1
do while n<7
desc[n]=space(20)
no_mot[n]=space(3)
capa[n]=space(3)
op_hour[n]=space(4)
n=n+1
enddo
```

```
define window cover1 from 1,1 to 6,70 double;
shadow
```

```
define popup fl_elec2 from 15,15 to 24,63;
shadow
```

```
define bar 2 of fl_elec2 prompt "                OPEN A NEW
FILE"
```

```
define bar 4 of fl_elec2 prompt "                DISPLAY NAMES OF
EXISTING FILES"
```

```
define bar 6 of fl_elec2 prompt "                RETRIEVE AN
EXISTING FILE"
```

```
define bar 8 of fl_elec2 prompt "                PRINT A
FILE"
```

```
on selection popup fl_elec2 do l_elec2
```

```
define popup fm_elec2 from 8,30 to 17,55;
prompt files like e2*.mem;
shadow
```

```
on selection popup fm_elec2 do m_elec2
```

```
activate popup fl_elec2
return
```

```
procedure l_elec2
do case
```

```

case bar()=2
  do elec2_nw
case bar()=4
  activate popup fm_elec2
case bar()=6
  do elec2_7
case bar()=8
  do elec2_9

endcase
return

procedure m_elec2
deactivate popup
return

procedure elec2_nw
do elec2_2
do elec2_4
do elec2_5
do elec2_8
release all except *p
clear
deactivate window cover
return

procedure elec2_2
activate window cover
@ 1,7 SAY "      ECO: REPLACE STANDARD BELTS WITH " +;
          "ENERGY EFFICIENT BELTS"
@ 2,1 TO 16,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
clear gets
return

procedure elec2_3
@ 3,3 SAY "Audit code:" GET a code
@ 3,33 SAY "File name: E2" GET f_name pict "XXXXXX"
@ 3,53 say ".MEM"
@ 3,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 4,3 SAY "Electricity consumption cost:"
@ 4,33 SAY "$" get e_costp
@ 4,42 SAY "/kWh"
@ 4,48 SAY "Demand cost:"
@ 4,61 SAY "$" get d_costp
@ 4,68 SAY "/kW-month"
@ 5,3 SAY "Average percent energy saved with EE belts:" get
pcentsav pict "99"
@ 5,49 say "%"
@ 6,3 SAY "Estimated motor efficiency:" get moteff pict "99"
@ 6,33 say "%"

```

```

@ 6,38 SAY "Estimated motor partload factor:" get partload
pict "99"
@ 6,73 say "% "
@ 8,3 say "System      Description      Number of      Motor
Operating hours"
@ 9,3 say " #
capacity (HP)      hours/yr      " @ 10,3 say " 1"
@ 10,12 get desc[1]
@ 10,37 get no_mot[1]
@ 10,51 get capa[1]
@ 10,67 get op_hour[1]
@ 11,3 say " 2"
@ 11,12 get desc[2]
@ 11,37 get no_mot[2]
@ 11,51 get capa[2]
@ 11,67 get op_hour[2]
@ 12,3 say " 3"
@ 12,12 get desc[3]
@ 12,37 get no_mot[3]
@ 12,51 get capa[3]
@ 12,67 get op_hour[3]
@ 13,3 say " 4"
@ 13,12 get desc[4]
@ 13,37 get no_mot[4]
@ 13,51 get capa[4]
@ 13,67 get op_hour[4]
@ 14,3 say " 5"
@ 14,12 get desc[5]
@ 14,37 get no_mot[5]
@ 14,51 get capa[5]
@ 14,67 get op_hour[5]
@ 15,3 say " 6"
@ 15,12 get desc[6]
@ 15,37 get no_mot[6]
@ 15,51 get capa[6]
@ 15,67 get op_hour[6]
read
return

```

```

procedure elec2_4
do while .t.
do elec2_3
an= " " -
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an)<>"N"
loop

```

```

endif
exit
enddo
return

```

```

procedure elec2_5

```

```

n=1
dm_svngs=0
kw_svngs=0
dm_svn=0
do while n<7
  dm_svn=val(no_mot[n])*val(capa[n])*100/moteff*partload/100d.7
  46 dm_svngs=dm_svngs+dm_svn
  kw_svngs=kw_svngs+round(dm_svn*val(op_hour[n]),0)
  n=n+1
enddo
mb_svngs=kw_svngs*3412/1000000
d_svngs=round((kw_svngs*val(e_costp))+(dm_svngs*12*val(d_cost
p)),0)

```

```

@ 17,1 TO 22,78 DOUBLE
@ 17,33 SAY " CALCULATIONS "

```

```

@ 18,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 18,26 SAY "kWh/yr"
@ 18,36 get mb_svngs picture "9999.99"
@ 18,44 SAY "MMBtu/yr"
@ 19,3 SAY "Dollar savings:"
@ 19,19 SAY "$" get d_svngs picture "999999"
@ 19,28 SAY "/yr"
@ 20,3 SAY "Implementation cost: " get im_cost
@ 21,3 SAY "Simple payback:" get p_back
clear gets
return

```

```

procedure elec2_7

```

```

move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: E2" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("e2"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "E2"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1

```

```

    return
endif
restore from "e2"+trim(mem_file) additive
do elec2_2
do elec2_3
do elec2_4
do elec2_5
do elec2_8
release all except *p
clear
deactivate window cover
return

procedure elec2_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
    return
endif
if upper(an)<>"N"
    if file ("e2"+trim(f_name)+".mem")=.t.
        move window cover1 to 18,5
        activate window cover1
        @ 1,20 say "FILE E2"+upper(trim(f_name))+".MEM ALREADY
        EXISTS"
        @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
        "!"
        read
        deactivate window cover1
        if upper (an)<>"N"
            delete file "E2"+trim(f_name)+".MEM"
            save to "e2"+trim(f_name)
        endif
    else
        save to "e2"+trim(f_name)
    endif
endif
return

procedure elec2_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: E2" get
mem_file pict "XXXXXX"
@ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
    return

```

```

endif
if file ("E2"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "E2"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
  restore from "E2"+trim(mem_file) additive
define popup fp_elec2 from 15,15 to 24,63;
shadow
define bar 4 of fp_elec2 prompt "                PRINT TO
PRINTER"
define bar 6 of fp_elec2 prompt "                PRINT TO DOS FILE
filename.txt"
on selection popup fp_elec2 do p_elec2
activate popup fp_elec2
return

procedure p_elec2
do case
  case bar()=4
    move window cover1 to 17,5
    activate window cover1
    @ 2,20 say "PLEASE WAIT.....PRINTING "
    set print on
    set console off
    do pg_elec2
    eject
    set print off
    set console on
    release all except *p
    deactivate window cover1
    return
  case bar()=6
    move window cover1 to 17,5
    activate window cover1
    @ 2,20 say "PLEASE WAIT.....PRINTING "
    set print to "E2"+trim(f_name)+".txt"
    set print on
    set console off
    do pg_elec2
    set print off
    set print to
    set console on
    release all except *p
    deactivate window cover1
    return
endcase
return

```

**** PROGRAM elec3.PRG

```
mem_file=space(6)
a_code=space(16)
f_name=space(6)
pcentsav=5
moteff=90
partload=80
dm_svngs=0
kw_svngs=0
mb_svngs=space(6)
d_svngs=space(6)
im_cost=0
p_back=0
```

```
declare desc[6],no_mot[6],capa[6],op_hour[6],v_imp[6]
n=1
do while n<7
desc[n]=space(20)
no_mot[n]=space(3)
capa[n]=space(3)
op_hour[n]=space(4)
v_imp[n]=0
n=n+1
enddo
```

```
define window cover1 from 1,1 to 6,70 double;
shadow
define window cover2 from 1,1 to 20,68 double;
shadow
```

```
define popup fl_elec3 from 15,15 to 24,63;
shadow
define bar 2 of fl_elec3 prompt "                OPEN A NEW
FILE"
define bar 4 of fl_elec3 prompt "                DISPLAY NAMES OF
EXISTING FILES"
define bar 6 of fl_elec3 prompt "                RETRIEVE AN
EXISTING FILE"
define bar 8 of fl_elec3 prompt "                PRINT A
FILE"
on selection popup fl_elec3 do l_elec3

define popup fm_elec3 from 8,30 to 17,55;
prompt files like e3*.mem;
shadow
on selection popup fm_elec3 do m_elec3

activate popup fl_elec3
return
```



```

procedure l_elec3
do case
  case bar()=2
    do elec3_nw
  case bar()=4
    activate popup fm_elec3
  case bar()=6
    do elec3_7
  case bar()=8
    do elec3_9

```

```

endcase
return

```

```

procedure m_elec3
deactivate popup
return

```

```

procedure elec3_nw
do elec3_2
do elec3_4
do elec3_4x
do elec3_4a
do elec3_5
do elec3_8
release all except *p
clear
deactivate window cover
return

```

```

procedure elec3_2
activate window cover
@ 1,7 say "          ECO: USE SYNCHRONOUS BELTS WITH SPROCKET
DRIVES"
@ 2,1 TO 16,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
clear gets
return

```

```

procedure elec3_3
@ 3,3 SAY "Audit code:" GET a code
@ 3,33 SAY "File name: E3" GET f_name pict "XXXXXX"
@ 3,53 say ".MEM"
@ 3,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 4,3 SAY "Electricity consumption cost:"
@ 4,33 SAY "$" get e_costp
@ 4,42 SAY "/kWh"
@ 4,48 SAY "Demand cost:"
@ 4,61 SAY "$" get d_costp
@ 4,68 SAY "/kW-month"

```

```

@ 5,3 SAY "Average % energy saved with synchronous belts and
sprocket drives:" get pcentsav pict "99" @ 5,73 say "%"
@ 6,3 SAY "Estimated motor efficiency:" get moteff pict "99"
@ 6,33 say "%"
@ 6,38 SAY "Estimated motor partload factor:" get partload
pict "99"

```

```

@ 6,73 say "%"
@ 8,3 say "System      Description      Number of      Motor
Operating hours"
@ 9,3 say " #          motors

```

```

capacity (HP)      hours/yr      "

```

```

@ 10,3 say " 1"
@ 10,12 get desc[1]
@ 10,37 get no_mot[1]
@ 10,51 get capa[1]
@ 10,67 get op_hour[1]
@ 11,3 say " 2"
@ 11,12 get desc[2]
@ 11,37 get no_mot[2]
@ 11,51 get capa[2]
@ 11,67 get op_hour[2]
@ 12,3 say " 3"
@ 12,12 get desc[3]
@ 12,37 get no_mot[3]
@ 12,51 get capa[3]
@ 12,67 get op_hour[3]
@ 13,3 say " 4"
@ 13,12 get desc[4]
@ 13,37 get no_mot[4]
@ 13,51 get capa[4]
@ 13,67 get op_hour[4]
@ 14,3 say " 5"
@ 14,12 get desc[5]
@ 14,37 get no_mot[5]
@ 14,51 get capa[5]
@ 14,67 get op_hour[5]
@ 15,3 say " 6"
@ 15,12 get desc[6]
@ 15,37 get no_mot[6]
@ 15,51 get capa[6]
@ 15,67 get op_hour[6]

```

```

read
return

```

```

procedure elec3_3a
move window cover2 to 5,5
activate window cover2
@ 1,3 say "      SYNCHRONOUS BELT DRIVES IMPLEMENTATION
COST      "

```

```

@ 2,3 say " -----"
-----"

```

```

@ 3,3 say " Description      Motor
Implementation "
@ 4,3 say "          capacity (HP)      cost

```

```

($)"
@ 5,3 say " -----
-----"
@ 6,7 get desc[1]
@ 6,33 get capa[1]
@ 6,46 get v_imp[1]
@ 7,7 get desc[2]
@ 7,33 get capa[2]
@ 7,46 get v_imp[2]
@ 8,7 get desc[3]
@ 8,33 get capa[3]
@ 8,46 get v_imp[3]
@ 9,7 get desc[4]
@ 9,33 get capa[4]
@ 9,46 get v_imp[4]
@ 10,7 get desc[5]
@ 10,33 get capa[5]
@ 10,46 get v_imp[5]
@ 11,7 get desc[6]
@ 11,33 get capa[6]
@ 11,46 get v_imp[6]
@12,3 say " -----
-----"
@13,3 say " Note: Default implementation cost is based on a
"
@14,3 say " reasonable estimate of $9.65 per motor
HP, based"
@15,3 say " on values obtained from vendor
consultation. "
read
return

```

```

procedure elec3_4
do while .t.
do elec3_3
an= " "
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an)<>"N"
loop
endif
exit
enddo
return

```

```

procedure elec3_4x

```

```

v_cost=9.65
n=1
do while n<7
  v_imp[n]=round(v_cost*val(capa[n]),0)
  n=n+1
enddo
return

procedure elec3_4a
do while .t.
do elec3_3a
an= " "
set colo to rg*/b+
@ 17,23 say " ANY CHANGES? (Y/N) " get an pict "!"
read
set colo to w+/b+
@ 17,23 say space(30)
set colo to w+/b+
if lastkey()=27
  deactivate window cover2
  return
endif
if upper(an)<>"N"
  loop
endif
deactivate window cover2
exit
enddo
return

procedure elec3_5
n=1
dm_svngs=0
kw_svngs=0
dm_svn=0
do while n<7
  dm_svn=val(no_mot[n])*val(capa[n])*100/moteff*partload/100d.7
  46 dm_svngs=dm_svngs+dm_svn
  kw_svngs=kw_svngs+round(dm_svn*val(op_hour[n]),0)
  n=n+1
enddo
mb_svngs=kw_svngs*3412/1000000
d_svngs=round((kw_svngs*val(e_costp))+(dm_svngs*12*val(d_cost
p)),0)
n=1
im_cost=0
do while n<7
  im_cost=im_cost+v_imp[n]*val(no_mot[n])
  n=n+1
enddo
p_back=round(im_cost/d_svngs,1)

@ 17,1 TO 22,78 DOUBLE
@ 17,33 SAY " CALCULATIONS "

```

```

@ 18,3 SAY "Energy savings:" get kw_svngs picture "999999"
@ 18,26 SAY "kWh/yr"
@ 18,36 get mb_svngs picture "9999.99"
@ 18,44 SAY "MMBtu/yr"
@ 19,3 SAY "Dollar savings:"
@ 19,19 SAY "$" get d_svngs picture "999999"
@ 19,28 SAY "/yr"
@ 20,3 SAY "Implementation cost: " get im_cost pict "999999"
@ 21,3 SAY "Simple payback:" get p_back pict "99.9"
clear gets
return

```

```

procedure elec3_7
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: E3" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
return
endif
if file ("e3"+trim(mem_file)+".mem")<>.t.
move window cover1 to 17,5
activate window cover1
an=" "
@ 1,20 say "E3"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
@ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
read
deactivate window cover1
return
endif
restore from "e3"+trim(mem_file) additive
do elec3_2
do elec3_3
do elec3_4
do elec3_4a
do elec3_5
do elec3_8
release all except *p
clear
deactivate window cover
return

```

```

procedure elec3_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+

```

```

@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
  return
endif
if upper(an) <> "N"
  if file ("e3"+trim(f_name)+".mem")=.t.
    move window cover1 to 18,5
    activate window cover1
    @ 1,20 say "FILE E3"+upper(trim(f_name))+".MEM ALREADY
    EXISTS"
    @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
    "!"
    read
    deactivate window cover1
    if upper (an) <> "N"
      delete file "E3"+trim(f_name)+".MEM"
      save to "e3"+trim(f_name)
    endif
  else
    save to "e3"+trim(f_name)
  endif
endif
return

procedure ELEC3_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: E3" get
mem_file pict "XXXXXX"
@ 2,61 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("E3"+trim(mem_file)+".mem") <> .t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "E3"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
restore from "E3"+trim(mem_file) additive
define popup fp_ELEC3 from 15,15 to 24,63;
shadow
define bar 4 of fp_ELEC3 prompt "          PRINT TO
PRINTER"
define bar 6 of fp_ELEC3 prompt "          PRINT TO DOS FILE
filename.txt"
on selection popup fp_ELEC3 do p_ELEC3

```

```
activate popup fp_ELEC3
return
```

```
procedure p_ELEC3
do case
case bar()=4
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print on
set console off
do pg_ELEC3
eject
set print off
set console on
release all except *p
deactivate window cover1
return
case bar()=6
move window cover1 to 17,5
activate window cover1
@ 2,20 say "PLEASE WAIT.....PRINTING "
set print to "E3"+trim(f_name)+".txt"
set print on
set console off
do pg_ELEC3
set print off
set print to
set console on
release all except *p
deactivate window cover1
return
endcase
return
```

**** PROGRAM boill.PRG

```
mem_file=space(6)
a_cōde=space(16)
f_name=space(6)
f_con=space(6)
save=0
ng_svngs=space(6)
mb_svngs=space(6)
d_svngs=space(6)
im_cost="NIL"
p_back="IMMEDIATE"
```

```
m_o2=0
m_temp=0
tI=0
ol=0
```

```
effl_2=0
effu_2=0
effl_1=0
effu_1=0
effl_u=0
effu_u=0
eff_2=0
eff_l=0
eff_u=0
sav_l=0
sav_u=0
sav=0
```

```
define window cover1 from 1,1 to 6,70 double;
shadow
```

```
define popup fl_boill from 15,15 to 24,63;
shadow
```

```
define bar 2 of fl_boill prompt "                OPEN A NEW
FILE"
```

```
define bar 4 of fl_boill prompt "                DISPLAY NAMES OF
EXISTING FILES"
```

```
define bar 6 of fl_boill prompt "                RETRIEVE AN
EXISTING FILE"
```

```
define bar 8 of fl_boill prompt "                PRINT A
FILE"
```

```
on selection popup fl_boill do l_boill
```

```
define popup fm_boill from 8,30 to 17,55;
prompt files like b1*.mem;
```

```
shadow
```

```
on selection popup fm_boill do m_boill
```



```
activate popup fl_boill
return
```

```
procedure l_boill
do case
  case bar()=2
    do boill_nw
  case bar()=4
    activate popup fm_boill
  case bar()=6
    do boill_7
  case bar()=8
    do boill_9
endcase
return
```

```
procedure m_boill
deactivate popup
return
```

```
procedure boill_nw
do boill_2
do boill_4
do boill_5
do boill_8
release all except *p
clear
deactivate window cover
return
```

```
procedure boill_2
activate window cover
@ 1,7 SAY "          ECO: REDUCE EXCESS BOILER
COMBUSTION" +;
      " AIR"
@ 2,1 TO 8,78 DOUBLE
@ 2,34 SAY " AUDIT DATA "
clear gets
return
```

```
procedure boill_3
@ 3,3 SAY "Audit code:" GET a code
@ 3,33 SAY "File name: E2" GET f_name pict "XXXXXX"
@ 3,53 say ".MEM"
@ 3,62 SAY "Date:" get datp picture "XX/XX/XX"
@ 4,3 SAY "Natural gas cost:"
@ 4,20 SAY "$" get n_costp
@ 4,29 SAY "/MCF"
@ 5,3 SAY "Boiler fuel consumption:" get f_con pict "99999"
```

```

@ 5,33 say " MCF/month"
@ 6,3 SAY "Percentage Oxygen in flue gas:" get m_o2 pict
"99.9" range 2,11
@ 6,38 say " %"
@ 7,3 SAY "Temperature of flue gas:" get m_temp pict "999"
range 250,850
@ 7,32 say "F"
read
return

```

```

procedure boill_4
do while .t.
do boill_3
an= " "
set colo to rg*/b+
@ 23,28 say "ANY CORRECTIONS? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,28 say space(40)
set colo to w+/b+
if lastkey()=27
return
endif
if upper(an) <> "N"
loop
endif
exit
enddo
return

```

```

procedure boill_5

```

```

if (m temp) < 350 .and. (m_temp) >= 250
t1=250
endif
if (m temp) < 450 .and. (m_temp) >= 350
t1=350
endif
if (m temp) < 550 .and. (m_temp) >= 450
t1=450
endif
if (m temp) < 650 .and. (m_temp) >= 550
t1=550
endif
if (m temp) < 750 .and. (m_temp) >= 650
t1=650
endif
if (m temp) < 850 .and. (m_temp) >= 750
t1=750
endif
if (m_temp)=850

```

```

t1=850
endif

if (m_o2) <3 .and. (m_o2) >=2
  ol=2
endif
if (m_o2) <4 .and. (m_o2) >=3
  ol=3
endif
if (m_o2) <5 .and. (m_o2) >=4
  ol=4
endif
if (m_o2) <6 .and. (m_o2) >=5
  ol=5
endif
if (m_o2) <7 .and. (m_o2) >=6
  ol=6
endif
if (m_o2) <8 .and. (m_o2) >=7
  ol=7
endif
if (m_o2) <9 .and. (m_o2) >=8
  ol=8
endif
if (m_o2) <10 .and. (m_o2) >=9
  ol=9
endif
if (m_o2) <11 .and. (m_o2) >=10
  ol=10
endif
if (m_o2) =11
  ol=11
endif

use boil1_d
locate for o2=2 .and. st_temp=t1
effl_2=eff
locate for o2=2 .and. st_temp=t1+100
effu_2=eff

locate for o2=ol .and. st_temp=t1
effl_1=eff
locate for o2=ol .and. st_temp=t1+100
effu_1=eff

locate for o2=ol+1 .and. st_temp=t1
effl_u=eff
locate for o2=ol+1 .and. st_temp=t1+100
effu_u=eff

close data

eff_2=effl_2+(effu_2-effl_2)/100*((m_temp)-t1)
eff_1=effl_1+(effu_1-effl_1)/100*((m_temp)-t1)

```

```

eff_u=effl_u+(eff_u-effl_u)/100*((m_temp)-t1)
sav_l=(eff_2-eff_l)/eff_2
sav_u=(eff_2-eff_u)/eff_2
if sav_l=1
  sav_l=0
endif
if sav_u=1
  sav_u=0
endif
sav=round(sav_l+(sav_u-sav_l)*((m_o2)-o1),3)
save=sav*100
ng_svngs=0
ng_svngs=round(val(f_con)*12*sav,0)
mb_svngs=ng_svngs
d_svngs=round(ng_svngs*val(n_costp),0)

```

```

@ 10,1 TO 18,78 DOUBLE
@ 10,33 SAY " CALCULATIONS "
@ 11,3 say "Expected percentage fuel savings by reducing"
@ 12,3 say " percentage of Oxygen in flue gas to 2% :" get
save pict "99.9"
@ 12,51 say "%"
@ 13,3 SAY "Natural gas savings:" get ng_svngs picture
"99999"
@ 13,30 SAY "MCF/yr"
@ 14,3 SAY "Energy savings:" get mb_svngs picture "99999"
@ 14,25 SAY "MMBtu/yr"
@ 15,3 SAY "Dollar savings:"
@ 15,19 SAY "$" get d_svngs picture "999999"
@ 15,28 SAY "/yr"
@ 16,3 SAY "Implementation cost: " get im_cost
@ 17,3 SAY "Simple payback:" get p_back
clear gets
return

```

```

procedure boill_7
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF EXISTING FILE: B1" get mem_file
pict "XXXXXX"
@ 2,56 say ".MEM"
read
deactivate window cover1
if lastkey()=27
  return
endif
if file ("b1"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=""
  @ 1,20 say "B1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"

```

```

    @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
    read
    deactivate window cover1
    return
endif
restore from "b1"+trim(mem_file) additive
do boill_2
do boill_3
do boill_4
do boill_5
do boill_8
release all except *p
clear
deactivate window cover
return

procedure boill_8
set colo to rg*/b+
an=" "
@ 23,30 say "SAVE FILE? (Y/N)" get an pict "!"
read
set colo to w+/b+
@ 23,30 say space(40)
set colo to w+/b+
if lastkey()=27
    return
endif
if upper(an)<>"N"
    if file ("b1"+trim(f_name)+".mem")=.t.
        move window cover1 to 18,5
        activate window cover1
        @ 1,20 say "FILE B1"+upper(trim(f_name))+".MEM ALREADY
        EXISTS"
        @ 2,20 say "DO YOU WANT TO OVERWRITE? (Y/N)?" get an pict
        "!"
        read
        deactivate window cover1
        if upper (an)<>"N"
            delete file "B1"+trim(f_name)+".MEM"
            save to "b1"+trim(f_name)
        endif
    else
        save to "b1"+trim(f_name)
    endif
endif
return

procedure BOILL_9
move window cover1 to 17,5
activate window cover1
@ 2,15 say "TYPE IN NAME OF FILE TO BE PRINTED: B1" get
mem file pict "XXXXXX" @ 2,61 say ".MEM"
read
deactivate window cover1

```

```

if lastkey()=27
  return
endif
if file ("B1"+trim(mem_file)+".mem")<>.t.
  move window cover1 to 17,5
  activate window cover1
  an=" "
  @ 1,20 say "B1"+upper(trim(mem_file))+".MEM DOES NOT EXIST"
  @ 2,20 say "PRESS ANY KEY TO CONTINUE" get an pict "!"
  read
  deactivate window cover1
  return
endif
restore from "B1"+trim(mem_file) additive
define popup fp_BOIL1 from 15,15 to 24,63;
shadow
define bar 4 of fp_BOIL1 prompt "                PRINT TO
PRINTER"
define bar 6 of fp_BOIL1 prompt "                PRINT TO DOS FILE
filename.txt"
on selection popup fp_BOIL1 do p_BOIL1
activate popup fp_BOIL1
return

procedure p_BOIL1
do case
case bar()=4
  move window cover1 to 17,5
  activate window cover1
  @ 2,20 say "PLEASE WAIT.....PRINTING "
  set print on
  set console off
  do pg_BOIL1
  eject
  set print off
  set console on
  release all except *p
  deactivate window cover1
  return
case bar()=6
  move window cover1 to 17,5
  activate window cover1
  @ 2,20 say "PLEASE WAIT.....PRINTING "
  set print to "B1"+trim(f_name)+".txt"
  set print on
  set console off
  do pg_BOIL1
  set print off
  set print to
  set console on
  release all except *p
  deactivate window cover1
  return
endcase

```

return

procedure pg_BOIL1

```
?  
?  
?  
?  
?  
? "      Audit code: "+trim(a_code)+"      Date: "+datp  
? "      File name: B1"+upper(trim(f_name))  
?  
? "      ECO: REDUCE EXCESS BOILER COMBUSTION  
AIR"  
?  
? "      RECOMMENDED ACTION"  
?  
? "      Oxygen is required for combustion of any fuel.  
This oxygen" is obtained from atmospheric air which is 21%  
? " oxygen by " volume. In theory it is possible to supply just  
? " the right " amount of air containing the oxygen required for  
? " complete " combustion of the fuel. In practice, however,  
? " getting the " air and fuel distributed and mixed evenly is  
? " virtually " impossible. For this reason an excess of  
combustion air is" recommended to ensure that all fuel burn  
? " completely. The " amount of excess air should be kept to a minimum  
? " (about 2% " oxygen) to avoid exhausting heated air from the  
stack. We " recommend that you monitor flue gas oxygen  
? " content using the" fuel efficiency analyzer and adjust air intake  
? " to maintain " 2% oxygen."  
?  
? "      DATA"  
?  
? "      Natural gas cost:          $" +n_costp+" /MCF"  
? "      Boiler fuel consumption:    "+f_con+"  
MCF/month"  
? "      Percentage Oxygen in flue gas: "+str(m_o2,8,2)+"  
%"  
? "      Temperature of flue gas:    "+str(m_temp,8)+  
" F"  
?  
? "      CALCULATIONS"  
?
```

```

? "      Expected percentage fuel savings by reducing"
? "      percentage of Oxygen in flue gas to 2% :
"+str(save,8,2)+" %"
?
? "      Natural gas savings:      "+str(ng_svngs,8,2)+"
MCF/yr"
?
? "      Energy savings      =      "+str(mb_svngs,8,2)+ "
MMBtu/yr"
?
? "      Dollar savings      = $ "+str(d_svngs,8)+ " /yr"
?
? "      Implementation cost =      Nil"
?
? "      Simple payback      =      Immediate"
? " "
return

```


APPENDIX C
PORTABLE PRINTER INFORMATION

Same Day Shipment • 30 Day Satisfaction Guarantee • Low Prices



GLOBAL

COMPUTER SUPPLIES

Canon BJ-10ex BUBBLE JET™ PRINTER

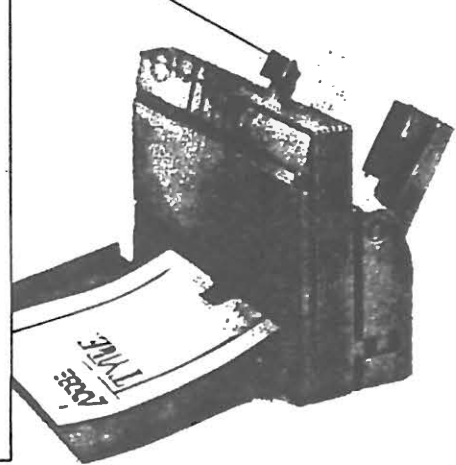
Now you can have letter quality precision wherever you go with the Canon BJ-10ex Bubble Jet™ Printer. Even with the optional rechargeable NiCad battery pack, the BJ-10ex weighs only 4.6 lbs. and measures 12 1/4" W x 8 1/2" D x 17 5/8" H, so it easily fits into your briefcase. The BJ-10ex is a full function desktop printer. The optional 30-sheet automatic sheet feeder is not only convenient, it stands to the side of the printer to save desk space.

BJ-10ex prints at a quiet 45dBA. It's so quiet that you can be printing right next to you and you will hardly know it's there. And Bubble Jet™ technology means that your output, text or graphics, will print out clean and crisp every time.

Canon BJ-10ex Specifications:

Printing Speed: High quality mode: 83 cps at 10 cps/100 cps at 12 cps/142 cps at 17 cps
 Resolution: 360 x 360 dots per inch
 Fonts: Courier, Pica, Elite, Roman, Sans Serif, Draft
 Emulation: IBM® Preprinter X24E, Canon BJ-150e, Epson® LQ-610 (LQ mode)
 Print Width: Maximum 8.0"
 Interface: Parallel (centronics)
 Noise Level: 45 dBA
 AC Power Adapter: 120V 30W included
 Weight: 4.6 lbs.
 Dimensions (W x D x H): 12 1/4" x 8 1/2" x 17 5/8"

Optional Auto-Sheet Feeder



Stock No.	Description	List Price	Global Low Price
C7092	Canon BJ-10ex Printer (Gray)	\$499.00	\$359.00
C7093	Canon BJ-10ex Printer (White)	499.00	359.00
C7094	Auto-Sheet Feeder (Gray)	89.00	65.00
C7095	Auto-Sheet Feeder (White)	89.00	65.00
C7096	Rechargeable NiCad Battery Pack	50.00	48.00
C9445	BJ Ink/Print Head Cartridge	30.00	25.00

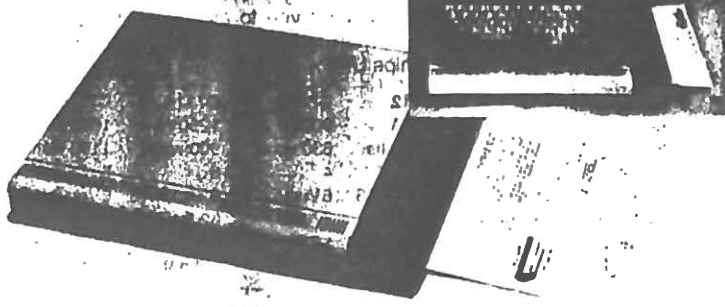
FOR SAME DAY SHIPMENT CALL 1-800-8-GLOBAL (that's 1-800-845-6226)

125

SEIKOSHA LT-20 NOTEBOOK PRINTER 24 pin notebook printer

The portable printer that combines a unique flatbed design and a built-in cassette tray for continuous printing of up to 50 sheets. The flatbed design also makes it the perfect stand for your laptop. Capable of both letter quality and multi-part forms handling, the LT-20 prints 180 characters per second (cps) in draft mode, and 60 cps in letter quality mode. Includes a detachable, 110 volt power cord, centronics parallel interface and a 2-yr. limited warranty. Take the LT-20 anywhere! Measuring 14 1/2" W x 11 1/4" D x 2" H and weighing only 5.9 lbs., it's thin and compact enough to fit neatly in your attache case. For total portability, order the optional Rechargeable Battery Pack and AC Battery Pack Charger (to add 8 additional fonts order the Font Expansion ROM Chip); prints up to 100 pages per battery charge.

Only **\$339⁰⁰**



Stock No.	Description	List Price	Global Price
C9449	LT-20 Notebook Printer	\$499.00	\$339.00
Also Available:			
C9456	Rechargeable Battery Pack	205.00	169.00
C7107	AC Battery Pack Charger	129.99	89.00
C9817	Font Expansion ROM Chip	29.99	24.00
C9454	LT-20 Printer Ribbon		15.00
C4415	6' Parallel Printer Cable		8.99

114 Four Global Locations • NY 1-516-625-6200 • CA 1-310-603-2286 • GA 1-404-339-9999 • IL 1-708-627-8800