EFFECT OF ENVIRONMENTAL FACTORS ON PAVEMENT DETERIORATION

FINAL REPORT
Volume II of II

PBA User's Manual
PBA Source Code
(Performance Based Approach)
(Version 1.0)

Gustavo Ordonez
Graduate Research Assistant

and

Ted S. Vinson
Principal Investigator

Department of Civil Engineering
Oregon State University
Corvallis, OR 97331

Transportation Research Institute
Report No. 88-31

NOVEMBER 1988

Document is available to the U.S. public through the
National Technical Information Service,
Springfield, Virginia 22161

Prepared for

Oregon Department of Transportation

Table of Contents

1. Introduction

PBA (Performance Based Approach) is a software package used to estimate the effect of environmental factors on pavement deterioration. The software is based on the Performance Based Approach as developed by Fwa and Sinha (1982). This approach uses pavement performance as predicted by performance models, and a relationship of pavement field performance to a quantification of level of routine maintenance, to evaluate pavement deterioration proportions of load and non-load related factors. The methodology of PBA is described in Volume I (Ordonez and Vinson, 1988); Volume II, the User's Guide, introduces you to PBA, explains the use of the software, and includes examples which will help you to become familiar with the software.

This User's Guide is your main reference to PBA. The Guide consists of the follwing sections:

Section 1: **Introduction** introduces the software package, describes the different modules that form the package, and lists the computer system requirements needed to execute PBA.

Section 2: **Main Menu** lists the main options found in PBA: Compute Theoretical-Performance PBA Index, Compute Field-Performance PBA Index, Estimate PBA Index for Zero-Level of Routine Maintenance, Compute Pavement Deterioration Proportions, Print a Data File, and End Execution of PBA.

Section 3: **Theoretical-Performance PBA Index** guides the user through the process of imputing the data to compute the PBA Index of a highway section based on a pavement performance model.

Section 4: **Field Performance PBA Index** explains the use of a field performance equation to compute the PBA Index of a pavement section.

Section 5: **User's Performance Equation** explains the entry of a customized performance equation to compute either the theoretical or the field PBA Index.

Section 6: **PBA Index for Zero-Level of Routine Maintenance** guides the user through the process of computing the PBA Index of a highway section which hasn't been subjected to routine maintenance; based on the data entered on the above options, and a measure of level of routine maintenance.

Section 7: **Pavement Deterioration Responsibilities** explains theo estimate of responsibilities of load and non-load related factors on pavement deterioration.

Section 8: **Error Messages** lists the different errors that are likely to occur when executing PBA.

Section 9: **Example** presents the steps required to enter the data used by PBA to compute the pavement deterioration proportions for a highway section.

1.1 System Requirements

PBA will execute on an IBM PC or compatible computer, under MS-DOS or PC-DOS, with or without a hard disk, with or without a math coprocessor, and with the following graphics cards: Hercules (black & white), CGA (color or black & white), and EGA (color or black & white). A minimum of 350K of RAM memory is required for the program and the working files.

A hard disk system is recommended; however, a system with two floppy disks is satisfactory if the PBA software is on drive A, and the data files are stored on the disk on drive B.

1.2 PBA Software

The BASIC source code of PBA was written with Borland's Turbo BASIC version 1.1. The source code presented in Appendix A, is formed by the following programs and files:

Source Code	**Description**
PBA.BAS	Main Menu, PBA index for field performance, PBA index for zero-level of routine maintenance.

PBASHARE.BAS	Routines to compute the pavement deterioration responsibilities.
PBAEQN.BAS	Routines to enter a customized performance equation.
PBAFFILE.BAS	File manager used by PBA.EXE, routines to input data.
PBAFEDIT.BAS	Routines to display screens, to edit data when using the input routine of PBAFFILE.BAS, and routines used to display context-sensitive help.
PBAASHTO.BAS	Routines to compute PBA index using the AASHTO equations.
PBAFILE.BAS	File manager and data input routine used by PBAASHTO.EXE.
PBAEDIT.BAS	Routines used to edit data being entered with PBAASHTO.EXE.
PBACAR.BAS	Routines used to enter vehicle types to predict ESALs using procedure described in Appendix D of the 1986 AASHTO Guide for the Design of Pavement Structures.
PBAHELP.BAS	Routines used to display the different input screens used by PBAASHTO.TBC, and to display context-sensitive help.
PBAASHTO.EXE	Compiled version of PBAASHTO.BAS.
HELP?.PBA	Help screens for the different options of PBA.
Q*.BIN	Inline Subroutines used by PBAHELP.BAS and PBAFEDIT.BAS to create the help-window, public domain software.

Q*.ASM	Assembler source of Q*.BIN inline subroutines, public domain software.

1.3 Installing PBA on your system

PBA can be executed on a two-floppy system or a system with a hard disk.

1.3.1 Installing PBA on a Floppy Disk System

The first thing you need to do is make back up copies of the original disk. For this:

- Get one new double sided, double density floppy disk.
- Boot up your computer.
- At the system prompt type the DOS command: **diskcopy A: B:**, press **ENTER**. if your system has only one floppy disk, you will see the message **Insert source diskette in drive A:**. Remove the disk on drive A (if any), and put the PBA disk into drive A.

  If your system has two floppy disks, then you will see the message: **Insert target diskette into drive B:**. Remove any diskette from drive B, and put the new diskette into drive B. Press **ENTER** to start copying the PBA (source) diskette on drive A to the new diskette (target) on drive B.

  If your system has only one floppy disk, you will be swapping disks in drive A. Hence, after reading the PBA disk, the computer will display a message asking you to insert the target (new diskette) into drive A. This can be repeated various times until copying is finished.

  Remove the original PBA diskette and store it in a safe place.

1.3.2 Setting PBA up on a Hard Disk

Create a directory called PBA (or any name you like) on your root directory. Assuming that the hard disk is drive C, at the system prompt type:

```
C:
CD\
md PBA
```

Next, put the PBA disk into drive A and type at the system prompt:

```
C:> copy a:*.* pba\*.*
```

Put the original PBA disk in a safe place.

2. Main Menu

At the system prompt type:

PBA

and then press **ENTER**. The introductory screen will be displayed after a few seconds. This screen shows the title of the research project, the sponsoring agency, and the developers of PBA. Press **ENTER** to continue executing PBA. The PBA Main Menu screen will be displayed.

PBA Main Menu
Oregon Department of Transportation
Highway Division
Performance Based Approach

1. **Compute Theoretical-Performance PBA Index**
2. **Compute Field-Performance PBA Index**
3. **PBA Index for Zero-Level of Routine Maintenance**
4. **Pavement Deterioration Proportions**
5. **Exit to DOS**

The menu screen is divided into three parts. The upper part displays the five options of PBA. Every time you call up the Main Menu, Option 1 will be highlighted. The middle part displays a message which gives a brief explanation of each of the five options. The bottom part of the screen shows the active keys, followed by a word which refers to the action that will be performed after pressing the key. For the Main Menu, the active keys are:

Key	**Action**
(down arrow key)	move the highlighted cell downward.

↑ (up arrow key)	move the highlighted cell upwards.
ENTER	selects the highlighted option.

Press ↓ (down arrow key) several times, to move the highlighted cell downward through the different options. A brief description of each option is displayed on the middle part of the screen. Also pressing ↑ (up arrow key) to move the options cell upwards.

2.1 Compute Theoretical-Performance PBA Index

The Performance Based Approach compares field pavement performance to predicted pavement performance. Total pavement deterioration is represented as the change in pavement condition from an ideal state to a condition state resulting from lack of routine maintenance. On the other hand, the change in pavement condition from the ideal state to that predicted by theoretical performance models represents the deterioration due only to load related factors. The ratio of these two changes will represent the deterioration responsibility of purely load related factors.

Option 1, Compute Theoretical-Performance PBA Index, will estimate the later change in pavement condition. This change is called the Theoretical-PBA Index herein. Refer to Section 3 of Volume I for further information.

2.2 Compute Field-Performance PBA Index

This option computes the change in pavement condition from an initial serviceability level to a level as measured in the field of a pavement section. This Field-PBA Index will be used in option 3 to estimate the total pavement deterioration as described above. For further information, refer to Sections 3.3 and 3.4 of Volume I.

2.3 PBA Index for Zero-Level of Routine Maintenance

The true total pavement deterioration of a highway section can be determined by knowing the Field-PBA Index of a number of pavement sections, and their respective quantification of level of routine maintenance. This option computes the PBA Index, which represents true total deterioration, by applying regression analysis to the data computed with option 2, and the data entered through this option. For further information refer to Section 3.4 of Volume I.

2.4 Pavement Deterioration Proportions

This option computes the pavement deterioration responsibilities based on the values computed with options 1 through 3. The output will be the pavement deterioration responsibility due to load-related factors, and the responsibility assigned to non-load related factors.

2.5 Exit to DOS

Execution of PBA can be finished by selecting option 6, this will return control to DOS. Termination of the program execution can also be done by pressing the F9 key when it is displayed in the bottom part of the screen.

3. Theoretical-Performance PBA Index

This option will estimate the change in pavement condition from an ideal state to that state predicted by a theoretical performance model. This change is named Theoretical-PBA Index. The methodology used for this is explained in Section 3 of Volume I (Ordonez and Vinson, 1988).

3.1 File Menu

The data used for estimating the Theoretical-PBA Index will be stored in a Random file with the extension **.DAT**. After choosing option 1, the File Menu Screen will be displayed. There are 4 options in this menu. For selecting one of these options, enter the option's number in the highlighted input cell and then press **ENTER**. For Example, to select option 3:

Enter a choice: 3
<press ENTER>

3.1.1 Create a New File

This option lets you create a brand new file to store the data used in computing the Theoretical-PBA Index, and the index also. To select this option, enter 1 on the highlighted input cell and then press ENTER:

Enter a choice: 1
<press ENTER>

After which, instructions to enter the file name will be displayed in the middle part of the screen. On the upper part of the screen the title:

New Data File

will be highlighted. Below this, in the middle part of your screen, the following instructions are displayed:

Enter the name of the data file to be created. You can add a disk drive address (eg. A:), or a directory path (eg. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

A highlighted cell is displayed following this message. By default, the data file is given the extension **.DAT**, therefore, do not add a file extension, or an error message will be displayed. Enter the name of the file and then press ENTER. For example:

C:\PBA\TEST
<press ENTER>

One of three things may happen:

1. The file name entered does not correspond to an already existent file, then, the data input screen described in Section 3.2 will be displayed.

2. The file name entered already exists. If this is the case, then two options are displayed on the bottom part of the screen:

   C:\PBA\TEST.DAT already exists.
   Do you want to overwrite/delete it (y/n) ?
   Do you want to add new information (y/n) ?

   and an input cell will be highlighted next to the overwrite/delete option. Then:

2.1 If you enter **Y**, the following message is displayed:

The data stored in C:\PBA\TEST.DAT
are to be deleted.
Enter 'y' to continue, 'n' to cancel

The input cell is now next to the word **cancel.** Entering a **Y** will erase the file and create a new file with the same name. The data previously stored in it will be lost. Entering a **N** will cancel deleting of the file and will return you to the file menu.

2.2 If you enter **N**, the input cell moves down next to the **new information** option. Entering **Y** will append the data to be entered in following sections, to the end of the data file, and the data input screen described in section 3.2 will be displayed. If you enter a **N**, you will return to the file menu, and you will have the choice of entering a new file name as previously described.

3. An error message will be displayed in the bottom part of the screen. For further information on the error message, refer to Section 8 of this User's Guide. Press **ESC** to return to the file menu.

3.1.2 Open an Existing File

This option opens an existing file and then appends the new data to the end of the file. To select this option enter 2 on the highlighted input cell and press ENTER:

Enter a choice: 2
<press ENTER>

Instructions to enter the file name will be displayed. In the upper part of the screen the title:

Open Data File

will be highlighted. Below this, in the middle part of the screen, the following message is displayed:

**Enter the name of the file to be opened
You can add a disk drive address (eg. A:) or a directory path (eg. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).**

A highlighted cell is displayed following this message. Do not enter a file extension; by default the file is given the extension **.DAT**. Enter the name of the file and then press ENTER; for example:

**C:\PBA\TEST
<press ENTER>**

One of two things may happen:

1. A file with that name is found, then the following message will be displayed on the bottom part of the screen:

   **C:\PBA\TEST.DAT already exists.
   Do you want to overwrite/delete it (y/n) ?
   Do you want to add new information (y/n) ?**

   An input cell will be highlighted next to the **overwrite/delete** option. Then:

   1.1 If you enter **Y**, the following message is displayed:

   **The data stored in C:\PBA\TEST.DAT
   are to be deleted.
   Enter 'y' to continue, 'n' to cancel**

   The input cell is now next to the word **cancel**. Entering **Y** will delete the file from the indicated path and create a new file with the same name. Entering **N** will cancel deleting of the file and will return you to the file menu.

   1.2 If you enter **N**, the input cell moves down to the new **information** option. Entering a **Y** will append the new data to be entered in following sections at the end of the file, and the data input screen described in section 3.2 will be displayed.

Entering **N** will return you to the file menu and you will have to enter a new file name as previously explained.

2. If a file with the indicated path and name can not be found, an error message will be displayed in the bottom part of your screen. Press **ESC** to return to the file menu, and enter a new file name. If a different error message is displayed, refer to Section 8 for further information.

3.1.3 View a Data File

This option lets you view the contents of a data file with the extension **.DAT**, which is a file that stores the data used to compute the Theoretical-PBA Index. To select this option enter a 3 on the highlighted input cell, and then press ENTER:

Enter a choice: 3
<press ENTER>

Instructions to enter the file name will be displayed. In the top part of the screen, the title:

View Data File

will be highlighted. Below this, in the middle part of the screen, the following message is displayed:

Enter the name of the file to be seen
You can add a disk drive address (eg. A:) or a directory path (eg. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

A highlighted cell is displayed following the message. By default, the file will be given the extension **.DAT**. Do not add a file extension or an error message will be displayed. Enter the file name, for example:

C:\PBA\TEST
<press ENTER>

One of two things may happen:

1. A file with the indicated path and name is found. The contents of the file will be displayed on the screen, one record at a time. The information displayed is:

   Record Number:
   Highway ID:
   PBA Index:
   District: **County:**
   Begin Mile Post: **End Mile Post:**
   Section EA: **EA/SJ:**
   Rehabilitation/Overlay Year: **Region:**

   Resilient Modulus: **Normal Deviate:**
   Standard Deviation: **Structural Number:**

   Dynamic Load: **Plate Radius:**
   Pavement Thickness: **Subgrade Poisson Ratio:**
   Field Structural Number: **Deflection Factor:**
   As Built Deflection: **Deflection Design:**

   Some of the above data may not show any information or values (ie. a blank space next to them). This depends on whether the AASHTO equations were used, or a user's performance equation was used to compute the Theoretical PBA Index. Refer to Section 3.2 for further information on the data displayed. To view the next record press **ENTER**. To quit viewing and return to the file menu press **ESC**.

2. If a file with the indicated path and name was not found, an error message will be displayed in the bottom part of the screen. Press **ESC** to return to the file menu, and try with a different file name. If a different error message is displayed, refer to Section 8 for further information.

3.1.4 Return to PBA Main Menu

Execution of PBA can be finished by selecting option 4. This will return you to the PBA Main Menu.

3.2 Enter Data for Highway Section

After a data file has been opened to store the information as described in sections 3.1.1 and 3.1.2, the input data screen for highway information is displayed. The title is shown in the upper part of the screen:

```
PREDICTED SERVICEABILITY MODULE
          DATA BASE
```

In the middle part of the screen, the following labels are displayed:

```
Identification:
    Highway             :
    District            :
    County              :
    Begin M.P.          :
    End M.P.            :
    Section EA          :
    EA/SJ               :
    Year                :
    Region              :

Performance Measure:
1.  User's Performance, AASHTO-PSI
2.  AASHTO Deflection

    Enter a choice:
```

The active keys are displayed at the bottom of the screen:

```
F1 : Help    F2 : Edit    F9 : End    ESC : Menu
```

A highlighted input-cell will be located next to the **Highway #** label, which will also be highlighted. A description of the meaning of these labels is given below. These descriptions are as used by the Oregon Department of Transportation, on its Pavement Management System.

3.2.1 Highway ID Data:

Information needed to properly identify a highway section is to be entered in this section. An example is provided to clarify the input procedure.

1. **Highway #:** To properly identify the highway section, enter any combination of number of letters, up to a maximum of 13 characters, and press **ENTER**. For example:

Highway # : **I-5 test**
<press ENTER>

2. **District:** Enter the name or number which identifies the district on which the highway section is located:

District : **2A**
<press ENTER>

3. **County :** Enter the name or number of the county where the highway section is located:

County : 26
<press ENTER>

4. **Begin M.P. :** Enter the name of the mile post which signals the beginning of the highway section:

Begin M.P. : 294.21
<press ENTER>

5. **End M.P. :** Enter the mile post which signals the ending of the highway section:

End M.P. : 299.56
<press ENTER>

6. **Section EA :** This is for specific purposes of ODOT:

Section EA : M0404414
<press ENTER>

7. **EA/SJ :** This is for specific purposes of ODOT:

EA/SJ : M6264430-001
<press ENTER>

8. **Year :** This refers to the year when the highway section was originally built, or the year when the last rehabilitation/ overlay job was undertaken in the highway section. For example,

if a highway section was built in 1963, and an overlay was placed on it on 1971, then enter the later year:

Year : 1971
<press ENTER>

9. **Region** : Enter the number which identifies the geographical region where the highway section is located:

Region : 4
<press ENTER>

After entering these data, refer to Section 3.2.2 for more information on the performance measure.

3.2.1.1 Editing

If you make a mistake when entering data, it can be edited by pressing the **F2** key. Instructions to edit data are displayed on the upper part of the screen:

↑ : vertical ← : horizontal ENTER : edit ESC : exit

and the data cell next to the **Highway** # label will be highlighted. A brief description of the active keys is given below:

↑ ↓ : vertical	Arrow keys are used to move the highlighted cell in a vertical direction. The data-cell will move downward every time one of these keys is pressed. When the data-cell reaches the last position of the input-cell

before the editing routine was called, the data-cell moves to the position next to the **Highway #** label. Use these keys to position the highlighted data-cell where you want to edit data.

← → : horizontal	These arrow keys move the data cell horizontally. These keys are not active when inputing Highway Identification Data.
ENTER : edit	When the highlighted data-cell is located on the data that you want to edit, press **ENTER** to input new data. After the key is pressed, a blank data-cell is displayed, and a blinking cursor displayed on it. Enter the new data, and press **ENTER** to finish editing. The highlighted cell shows the new data. Use the arrow keys to move the data-cell to a different position.
ESC : exit	After you have corrected any mistake, press **ESC** to continue executing PBA. This will position the blinking cursor back to where it was when editing was called.

For example, a mistake was made when entering the data for **Section EA** above:

```
Highway #            : I-5 test
District             : 2A
County               : 26
Begin M.P.           : 294.21
End M.P.             : 299.56
Section EA           : M0403414
EA/SJ                : M6264_
```

Instead of M0403414 you should enter M0404414. Press **F2**. The editing screen will be displayed on the upper part of the screen. The cell next to the **Highway #** label is highlighted:

```
Highway #            : I-5 test
```

Press the arrow keys until the highlighted data-cell is located on the **Section EA** value. Press **ENTER** to clear the data-cell and enter the new value:

Section EA : **M0404414**
<press ENTER>

Press **ESC** to exit editing and return to continue executing PBA. The blinking cursor is next to the 4 in the EA/SJ data cell as before.

3.2.2 Performance Measure

The Performance Based Approach uses theoretical and field pavement performance curves to determine deterioration responsibilities. PBA was developed to work with either a theoretical or a customized performance model. The theoretical models used are the AASHTO design equation 1.2.1 (AASHTO, 1986), as a function of Present Serviceability Index; and, equation PP.26 for deflection as function of pavement structural data and dynamic NDT deflection device information (AASHTO, 1986). A customized performance model can be entered into PBA by breaking the equation down in sub-equations (up to 5 sub-equations), and by defining the relationship between them. The procedure to enter a user's performance model is provided in section 5.

After you have entered the highway identification data, the highlighted input-cell is positioned next to the label **Enter a**

choice: in the bottom part of the screen. Select a model for a theoretical estimate of performance:

1 for AASHTO Equation 1.2.1
1 for User's Performance Equation
2 for AASHTO Equation PP.26

3.2.2.1 User's Performance, AASHTO-PSI

If you choose option 1, the default performance model is AASHTO design equation 1.2.1, as function of Equivalent Single Axle Loads (herein ESALs). If you would like to use a customized performance model, refer to Section 5 for further information on entering a performance equation in PBA. After entering 1, the bottom part of the screen looks like:

Independent Performance Variable (X axis):
Predicted Traffic (y/n):
Manual Input (y/n):

Independent Performance Variable refers to the variable the performance model is function of. For example, AASHTO design equation 1.2.1 was rearranged so that PSI is a function of ESALs.

If the performance model selected is the AASHTO Design equation 1.2.1, ESALs can be entered by any of two procedures:

1. Predicted: ESALs are estimated as explained in Appendix D of the 1986 AASHTO Guide for the Design of Pavement Structures (AASHTO, 1986). For instructions on how to enter the data refer to Section 3.3. To select this option enter a Y:

Predicted Traffic (y/n): y
<press ENTER>

2. Manually: If traffic measurements have been taken during the pavement life, these can be input manually. To choose this option enter a **N** on the **Predicted Traffic** option, and **Y** in the **Manual Input** option:

   Predicted Traffic (y/n): n
   <press ENTER>
   Manual Input (y/n): y
   <press ENTER>

   Refer to section 3.4 for further instructions on how to input these values.

If neither predicted or manual option is chosen, the user will be returned to the Performance Measure options:

Predicted Traffic (y/n): n
<press ENTER>
Manual Input (y/n): n
<press ENTER>

The following menu is displayed in the bottom part of the screen:

Performance Measure:
1. User's Performance, AASTHO-PSI
2. AASHTO Deflection

If the performance model is to be entered by the user, and the independent variable is ESALs, then the values for the variable can be either predicted or entered manually, as described above. If the independent variable is another type of measurement, then its values should be entered manually, as described in Section 3.4.

3.2.2.2 AASHTO Deflection

The default performance model for option 2 is AASHTO Equation PP.26 (AASHTO, 1986). This equation is a function of pavement structural data and dynamic NDT deflection device information. To select this option, enter a 2:

Enter a choice: 2
<press ENTER>

The screen to select a performance equation will be displayed. Refer to Section 3.5 to enter data for computing theoretical pavement deflection.

3.2.3 Keys

The active keys when entering the highway identification data and their functions, are described below:

F1 : Help	Displays a context-sensitive help screen. Parts of this document, and of Volume I will be displayed for quick information on using PBA.
F2 : Edit	Calls the editing routine as previously described in Section 3.2.1.10.
F9 : End	Finishes the execution of PBA and returns control to DOS. If data were entered, then they will be lost.
ESC : File Menu	Returns you to the File Menu described in Section 3.1.

3.3 Predicted Traffic

PBA estimates ESALs, based on the guidelines provided in Appendix D of the 1986 AASHTO Guide for the Design of Pavement Structures. The information needed to compute traffic is based on vehicle types, current traffic, growth factors, and ESAL factors.

After selecting this option, as explained in Section 3.2.2.1, a file menu will be displayed. The title:

Vehicle Types

is displayed on the top part of the screen, and the file options in the middle part:

1. AASHTO Vehicle Types
2. Open an Existing Vehicle-Types File
3. Create a Vehicle-Types File
4. Return to Manual Traffic
Enter a Choice:

To select any of these options, enter the number in the left hand side. Each of these options is described below.

3.3.1 AASHTO Vehicle Types

This option lets you use the vehicle types used in Appendix D of the AASHTO Guide to estimate ESALs. These vehicle types are listed below:

Passenger Cars
Buses
Panel and Pickup Trucks

Other 2-Axle/4-Tire Trucks
2-Axle/6-Tire Trucks
3 or More Axle Trucks
3 Axle Tractor Semi-Trailers
4 Axle Tractor Semi-Trailers
5+ Axle Tractor Semi-Trailers
5 Axle Double Trailers
6+ Axle Double Trailers
3 Axle Truck-Trailers
4 Axle Truck-Trailers
5+ Axle Truck-Trailers

To use a different set of vehicle types, refer to Section 3.3.3. After selecting option 1, the data-input screen described in Section 3.3.5 is displayed.

3.3.2 Existing Vehicle-Types File

Select this option if you want to use a set of vehicle types that you created before. To choose this option enter a 2:

Enter a choice: 2
<press ENTER>

The instructions to enter the name of the file that stores the vehicle types needed are displayed; on the top part of the screen, the title:

Open Vehicle-Type File

will be displayed below with the message:

Enter the name of the Vehicle-Type file to be opened You can add a disk drive address (eg. A:) or a directory path (eg. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

Enter the name of the file to be opened in the highlighted input-cell, and press **ENTER**. For example:

C:\PBA\TEST
<press ENTER>

One of two things may happen:

1. The file name corresponds to an existing Vehicle-Type file. The vehicle types will be retrieved, and the data-input screen described in Section 3.5 displayed. Refer to this section for instructions on how to enter the data needed to compute ESALs.

2. A file with the given name is not found. Then, an error message will be displayed in the bottom of the screen, press ESC to continue. You will return to the file menu described in Section 3.3.

3.3.3 New Vehicle-Types File

Use this option to create a file that stores a set of vehicle types used to compute ESALs. To select this option enter a 3:

Enter a choice: 3
<press ENTER>

The following title will be highlighted on the upper part of the screen:

New Vehicle-Type File

Instructions to enter the name of the file to be created are displayed on the middle part of the screen:

Enter the name of the Vehicle-Type file to be created You can use a disk drive address (eg. A:) or a directory path (eg. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

Type the name of the file in the highlighted input-cell and then press **ENTER**. The file will be given the extension **.CAR** by default. (If you add a file extension, an error message will be displayed on the bottom part of the screen.) For example:

C:\PBA\TEST
<press ENTER>

One of three things may happen:

1. The name entered does not correspond to an existing file. Then, the input-screen to enter vehicle types will be displayed. Refer to Section 3.3.3.1 for instructions on entering the vehicle types.

2. A file with the given name already exists. The following message is displayed on the middle of the screen:

   C:\PBA\TEST.CAR
   already exists.
   Do you want to overwrite/delete it (y/n) ?

   and an input-cell is highlighted next to the **overwrite/delete** option.

   If you enter Y, the following message is displayed:

   The data stored in C:\PBA\TEST.CAR
   are to be deleted.
   Enter 'y' to continue, 'n' to cancel

   the input-cell is now next to the word **cancel**. Entering a **Y** will erase the file, and create a new file with the same name, and the input-screen described in Section 3.5 displayed. The data previously stored in it will be lost. Entering **N** will cancel erasing the existing file, and will return you to the file menu described in Section 3.3.

If you enter **N**, the existing file is left unchanged, and you will return to the file menu described in Section 3.3.

3. If an error message is displayed in the bottom part of the screen, refer to Section 8 for more information.

3.3.4 Return to Manual Traffic

If you wish to enter ESALs manually, enter a 4. This will display the data input-screen described in Section 3.4. Refer to this section for instructions on entering ESALs.

3.3.5 Enter Vehicle Types

After you have successfully created a file to store the vehicle types, the data input-screen will be displayed. On the top of the screen, a title and a label requesting the number of vehicles are displayed:

```
      VEHICLE TYPES
Input the number of vehicle types
(maximum of 50 vehicle types)   :
```

A highlighted input-cell is displayed next to the last label. Enter the number of vehicle types to be input, from 1 to 50. (If you enter a number outside this range, the label **(maximum of 50 vehicle types)** will blink, and the input cell will clear.) For example:

```
(maximum of 50 vehicle types)  : 5
<press ENTER>
```

In the middle part of the screen, the tile:

Enter the Vehicle Types (28 characters maximum)

is displayed. Below this, a number of input-cells are displayed, with a number to their left representing the vehicle type being entered. The input-cell for the first data is highlighted, and the cursor is positioned at the beginning of the cell. For example:

Passenger Cars
<press ENTER>
Panel and Pickup Trucks
<press ENTER>
3 or More Axle Trucks
<press ENTER>
5 Axle Double Trailers
<press ENTER>
5+ Axle Truck-Trailers
<press ENTER>

If the number of vehicle types is greater than 10, then two or more screens will be needed to enter the data. After entering the vehicle types, the data input-screen to enter the data needed to compute ESALs is displayed. Refer to Section 3.3.6 for instructions on how to enter these data.

3.3.5.1 Editing Vehicle Types

If you made a mistake when entering the vehicle types, then press **F6** to edit the mistake. This will display editing instructions on the upper part of the screen:

↑↓ : vertical **ENTER : edit** **ESC : exit**

The data-cell next to the number 1 will be highlighted. A brief description of these active keys is given below:

↑↓ : vertical	Arrow keys are used to move the highlighted cell in a vertical direction. The data-cell will move downward every time one of these keys is pressed. When the data cell reaches the last position of the input-cell before editing was called, the data-cell moves next to the number 1. Use these keys to position the highlighted cell when you want to edit data.
ENTER : edit	When the highlighted data-cell is located on the data that you want to edit, press **ENTER** to input new data. A blank data-cell is displayed and a blinking cursor positioned at the beginning of it. Enter the new vehicle type, and then press **ENTER** to finish editing. The highlighted cell shows the new vehicle type. Use the arrow keys to move the data-cell to a new position.
ESC : exit	After you have edited a mistake, press **ESC** to continue executing PBA. This will position the blinking cursor back where it was before editing was called.

For example, a mistake was made when entering the second vehicle type above, and half of the third type was already typed when the mistake was seen:

1. Passenger Cars
2. Panel and Pick Trucks
3. 3 or More Ax_

Instead of **Pick** you should enter **Pickup.** Press **F6**, the editing screen will be displayed. The cell for the first vehicle type is highlighted. Press the ↑ or ↓ arrow key once, this positions the highlighted cell on the **Panel and Pick Trucks** vehicle type. Press **ENTER**, a blank data-cell is displayed. Type in the new vehicle type:

Panel and Pickup Trucks
<press ENTER>

Press **ESC** to exit editing. The blinking cursor is positioned next to the **x** in the third vehicle type. It is important to note that editing can be done for the vehicle types on the current data input-screen. Vehicle types that have been entered on previous screens can not be edited.

3.3.6 Traffic Data - AASHTO Method

To estimate ESALs for a design period, PBA uses the method described in Appendix D of the AASHTO Guide (1986). This method predicts traffic as Equivalent Single Axle Loads (ESALs), based on vehicle type information, traffic counts, growth factors, and ESAL factors. For further information refer to the AASHTO Guide. ESALs are computed for a design period, which begins the year you entered in the Highway Identification data in Section 3.2, and ends during the current year.

After you have successfully created a set of vehicle types, the data input-screen to enter the above data is displayed. The titles:

TRAFFIC DATA
AASHTO Method
Worksheet for calculating 18-kip ESAL applications

are displayed on the upper part of your screen. Below these, the **Highway ID** data entered before is displayed. On the middle part of the screen, a data frame with headings:

Vehicle Type	current traffic	growth rate	ESAL factor

is displayed. The vehicle types previously entered or retrieved are displayed on the **Vehicle Type** column. A highlighted input-cell is printed on the **current traffic** column of the first vehicle type. The following active keys are displayed on the bottom part of the screen:

F1 : Help **F6 : Edit** **ESC : Quit** **F9 : End**

For example, use the following values for the vehicle types entered before. (If you make a mistake, refer to Section 3.3.6.1 for instructions on editing mistakes.) Enter the current traffic value for Passenger Cars:

5925
<press ENTER>

The highlighted cell moves to the growth rate column. Enter:

4
<press ENTER>

The input-cell is positioned now in the ESAL factor column, Enter:

0.0008
<press ENTER>

The process repeats for the remaining vehicle types. Enter:

Panel and Pickup Trucks:
1135
4
0.0122

3 or More Axle Trucks:
34
4
0.1303

5 Axle Double Trailers:
1880
6
2.3719

5+ Axle Truck Trailers:
125
6
0.5317

After you enter the data for the last vehicle type, one of two things may happen:

1. The computer remains on the data input-screen for a few seconds. The ESALs are being computed. After this, the performance equation menu, described in Section 3.5 is displayed.
2. A small window is displayed on the bottom part of the screen asking you to enter the construction/

rehabilitation year. If you did not enter a value for the **Year** data in Section 3.2, PBA requests the year to compute ESALs for the design period. Assuming that one did not enter this value on the example of Section 3.2, enter:

1971
<press ENTER>

The data input-screen might remain displayed for a few seconds during which ESALs are being computed. The performance equation menu, described in Section 3.5, will be displayed next.

3.3.6.1 Edit Traffic Data

To edit mistakes you might make during the process of entering the data for predicting traffic, press the **F6** key. This will display an editing screen on the upper part of the screen. The active keys for editing are explained in Section 3.3.5.1, however, the ← → arrow keys are not, they are defined below:

← → : lateral	Arrow keys are used to move the highlighted cell in a lateral direction. The data-cell will move to the column to the right every time one of these keys is pressed. When the data cell reaches the last position of the input-cell before editing was called, or the last column, the data-cell moves to the first column of the current row. Use these keys to position the highlighted cell where you want to edit data.

The bottom of the screen, you are asked if you want to switch to a manual way of entering traffic data:

Switch to MANUAL TRAFFIC input (y/n) ?

A highlighted input-cell is displayed next to this option. If you press the **Y** key, the data input-screen for manual input is displayed, refer to Section 3.4 for instructions on entering traffic manually. If your press the **N** key, the data-cell on the **current traffic** column of the first vehicle type is highlighted.

For example, suppose you entered a 4 instead of 6 for the growth rate of the 5 Axle Double Trailers type. And you realized this when you were entering the ESAL factor for the 5+ Axle Truck Trailers. Then, press **F6**, and press the **N** key to answer the switch option:

<press> F6
<press> N

Use the arrow keys to position the highlighted data-cell in the growth rate column of the 5 Axle Double Trailers vehicle type. Press **ENTER**, a highlighted blank data-cell is displayed, and a blinking cursor positioned at the beginning of it. Enter the new value:

6
<press ENTER>

The highlighted data-cell now displays the new value. To edit other data, use the arrow keys to reposition the data-cell, and follow the above procedure. Press **ESC** after editing the mistakes. This will position the blinking cursor back on the ESAL factor column of the 5+ Axle Truck Trailers.

3.3.6.2 Keys

The active keys during the process of entering the traffic data are:

F1 : Help	This will display a help screen on the middle part of the main screen. Parts of this document, or of Volume I, will be displayed for a quick reference.
F6: Edit	Calls the editing routine described in section 3.3.6.1.
ESC : Quit	This key terminates the process of predicting traffic, and will return you to a manual way of inputing the data.
F9 : End	This key terminates the execution of PBA. The data entered until this point will be lost.

3.4 Independent Performance Variable; Manual Input

If traffic counts, or any other measure, have been made during the life of the pavement structure, these can be entered manually for estimation of PBA Index. To select this option, enter **N** for predicted traffic option, and **Y** for the independent variable option:

Predicted Traffic (y/n) : N
<press ENTER>

Manual Input (y/n) : Y
<press ENTER>

The input-screen for entering data manually is displayed. Depending on the type of data you are entering, refer to one of the sections below.

3.4.1 Independent Performance Variable

If you are entering traffic data manually, or any other measure, to be used as the independent variable for performance computation, you will be asked to enter data to identify the variable. The following title is displayed:

Prediction Values
Independent Variable of Predicted Performance

below which, input-cells for ID data are highlighted:

Enter the variable name and units:
Name:
Units:
Variable ID:

For example, enter ESAL's to be used in the following sections to predict Present Serviceability Index; the cursor is positioned in the highlighted cell for the **Name** data:

Name : Traffic
<press ENTER>
Units : ESAL
<press ENTER>
Variable ID : ESAL
<press ENTER>

A data-input screen will be displayed on the bottom part of the screen, with the following headings:

Highway ID : (data entered in section 3.2)
Year (variable ID entered above)

You will be requested to enter the number of measurements to be used. The message:

```
Input the number of measurements
(maximum of 10 measurements) :
```

will be displayed in the upper part of the screen. A highlighted input cell is placed next to the **(maximum of 10 measurements)** note. You can enter a number between 1 and 10. (Any number outside this range will not be accepted, and the above message will blink to remind you of this.) For example, enter 4 measurements to be used in the following sections:

```
(maximum of 10 measurements) : 4
<press ENTER>
```

An equal number of input-cells will be highlighted for each column in the data frame. Refer to Section 3.4.3 for additional information on how to enter the data.

3.4.2 User's Performance Equation; Theoretical PBA Index

If the performance model, used to compute the theoretical PBA Index, is being entered by the user, then data are to be entered manually for some of the variables forming the model. The data input-screen is displayed, with the following information shown on the top part of the screen:

```
Highway ID   :
Equation ID  :
Variable     :
No. Variable :
```

The data frame is displayed on the bottom part of the screen, with headings:

Year **Measure**

You will be requested to enter the number of measurements to be used. The message:

Input the number of measurements
(maximum of 10 measurements) :

is displayed in the upper part of the screen. A highlighted input-cell is placed next to the **(maximum of 10 measurements)** note. You can enter a number between 1 and 10. (Any number outside this range will not be accepted, and the above message will blink to remind you of that.) For example:

(maximum of 10 measurements) : 4
<press ENTER>

An equal number of input-cells will be highlighted for each column in the data frame. Refer to section 3.4.3 for more information on how to enter the data.

3.4.3 Manual Data Input

After you have entered the number of measurements, the input-cells are highlighted in the data frame. The cell at the first row of the first column is highlighted in a different color, and a blinking cursor positioned at the beginning of this cell. Enter the data and then press **ENTER**, this will move the data-cell to the next position where data is to be input. For example, enter the following ESAL values to continue with the example of Section 3.4.1:

1976	**<press ENTER>**
10003000	**<press ENTER>**
1980	**<press ENTER>**
23000000	**<press ENTER>**
1984	**<press ENTER>**
35065000	**<press ENTER>**
1987	**<press ENTER>**
46000000	**<press ENTER>**

If you make a mistake, refer to Section 3.4.4 for information on how to edit data. After you have entered the values, the performance equation menu, described in Section 3.5, will be displayed.

3.4.4 Edit Data for Manual Input

To edit mistakes made when entering the values manually, press F4 to call the editing routine. This will display an editing screen on the upper part of the input-screen:

↑↓ : vertical   ←→ : lateral   ENTER : edit   ESC : exit

A description of these active keys is given in Sections 3.3.5.1 and 3.3.6.1.

For example, assume that a mistake was made when entering the ESAL value for the 1980 year. Instead of 23000000, we entered 16500000. Press F4, in the bottom part of the screen and you will be asked if you wish to switch to a predictive way of estimating traffic:

Switch to PREDICTED TRAFFIC (y/n) ?

A highlighted cell is displayed next to this option. If you press **Y** the menu for choosing a vehicle-types file, described in Section 3.3, is displayed. If you press **N**, the data-cell on the number of measurements is highlighted. At this point you can do one of two things:

1. Increase the number of measurements. When the data-cell is on the number of measurements option, press **ENTER**. This will clear the data-cell. Then, enter the new number of measurements and press **ENTER**. If the new value is greater than the old number of measurements, then more input-cells will be highlighted in the data frame for each column. If the new value is equal to or less than the old value, nothing will happen. If you want to decrease the number of measurements, just enter a value of zero, or press **ENTER** for each of the data beyond the number of measurements you want to use. For Example, if you entered 7 for the number of measurements, but only wanted 4, enter zeros for the data-cells 5,6 and 7, or press **ENTER**, without any input, when the input-cell is positioned on these data-cells.

2. Edit Data. Use the arrow keys to position the data-cell where you want to edit data. For our example, place the data-cell on the ESAL column of the 1980 year. Press **ENTER**, this clears the data-cell, enter the right value:

   23000000
   <press ENTER>

   the new value is displayed. Use the arrow keys to position the data-cell if you have to correct more data. Press **ESC** to exit the editing routine, and go back to enter the remaining data.

3.4.5 Keys

The active keys during the process of manually entering data are:

F1 : help	This will display a help screen on the middle part of the screen. Parts of this document, or of Volume I, will be displayed for quick reference.
F4 : edit	Calls the editing routine described in Section 3.4.4.
ESC : quit	This will terminate the process of entering data and will return you to the identification data screen.
F9 : End	This key terminates the execution of PBA. The data entered until this point will be lost.

3.5 Performance Equation

After you have entered highway identification data, selected a performance measurement, and entered data for the independent variable of the performance equation, you will need to select a performance model to use in computing the theoretical PBA Index. There are two options: first, use the default AASHTO equations for PSI and Deflection; and, second, use a customized performance model. These options are discussed in the following sections. The performance model menu will be displayed. The title:

Performance Equation

is displayed on the upper part of the menu. Below this, the following options are shown:

1. **AASHTO Equations for Flexible Pavements.**
   Equations 1.2.1 & PP.26 of 1986 AASHTO Design Guide.
2. **User-Defined Performance Equation**
3. **Cancel Data for Current Section**

   Enter a choice:

To select an option, type the number of the option in the highlighted input-cell, and press the **ENTER** key. Refer to the following sections for further information.

3.5.1 AASHTO Equations

PBA uses equation 1.2.1, design equation for flexible pavements; and, equation PP.26 used to predict Structural Number for a particular Non Destructive Testing Device (AASHTO, 1986). These equations were rearranged so that equation 1.2.1 yields PSI as a function of ESALs and pavement structural data; and, equation PP.26 estimates deflection as a function of pavement structural data. The data needed to use these equations is described in sections 3.5.1.1 and 3.5.1.2.

3.5.1.1 Pavement Structural Data

After you select option 1 of the Performance Equation Menu, you will be asked to enter the pavement structural data needed in the AASHTO equations. An input-screen will be displayed with the highway identification data that were entered in Section 3.2 shown on the upper part of the screen:

Highway ID : (identification data)

In the middle of the screen, labels for the different data are displayed:

Roadbed Resilient Modulus (Mr):
Select a choice:
1. Mr value
2. CBR correlation
3. R-value correlation

Reliability (R, percent) =
<default: 0.95>

Standard Deviation (So) =
<default: 0.45>

A highlighted input-cell is displayed next to the **Select a choice:** label. The active keys are displayed on the bottom part of the screen:

F1 : Help   F8 : Edit   ESC : Quit   F9 : End

There are three options to input the value for resilient modulus:

1. If you know the value for Mr, then choose option 1.

   Select a choice: 1
   <press ENTER>

   This will highlight an input-cell next to the **1. Mr value** label. Type in the value and then press the **ENTER** key:

   1. Mr value   30000
   <press ENTER>

   The highlighted input-cell moves to the **Reliability (R, percent)** label. If you wish to change your choice for Mr, refer to Section 3.5.1.2, Editing Structural Data, for instructions.

2. If you would like to use equation 1.5.1 of the 1986 AASHTO Guide to compute Mr based on a CBR value, choose option 2. An input-cell is positioned next to the **2. CBR correlation** label. Enter a value for CBR and press **ENTER**. The resultant Mr value will be displayed next to the CBR

value:

2. CBR correlation : 20
<press ENTER>

2. CBR correlation : 20 Mr = 30000

The input-cell moves to the **Reliability (R, percent)** label. If you wish to change your choice for Mr, refer to Section 3.5.1.2, Editing Structural Data, for instructions.

3. To compute Mr using equation 1.5.3 of the 1986 AASHTO Guide, based on R-value, select option 3. The input-cell is positioned next to the **3. R-value correlation** label. Enter a value for R, and press the **ENTER** key. The resultant Mr will be displayed next to the R-value:

   3. R-value correlation : 40
   <press ENTER>

   3. R-value correlation : 40 Mr = 23200

   The input-cell moves down to the **Reliability (R, percent)** label. If you wish to edit data, refer to Section 3.5.1.2, Editing Structural Data.

After you have entered the resilient modulus, the input-cell moves next to the **Reliability** label. Enter a value for reliability. PBA will determine the value of the Standard Normal Deviate, from Table 4.1 of the AASHTO Guide, using this reliability value. You can enter the reliability value using one of two options:

1. If you don't enter a reliability value, PBA uses a default of 95 percent. Press **ENTER** and the default value for Standard Normal Deviate is displayed next to the reliability value:

   Reliability (R, percent) =
   <default: 0.95>
   <press ENTER>

Reliability (R, percent) = 95   Zr = -1.645
<default: 0.95>

The input-cell moves next to the **Standard Deviation** label. If you want to edit any data, refer to Section 3.5.1.2, Editing Structural Data.

2. Enter a value for Reliability, and then press **ENTER.** The value for Standard Normal Deviate is displayed next to the reliability value:

   Reliability (R, percent) = 90
   <default: 0.95>
   <press ENTER>

   Reliability (R, percent) = 90   Zr = -1.282
   <default: 0.95>

   The input-cell moves down to the **Standard Deviation** label. To edit data, refer to Section 3.5.1.2, Editing Structural Data.

The input-cell is now next to the **Standard Deviation (So)** label. For reliability, you can either use a default value of 0.45, or enter a different value for standard deviation. To use the default value, press the **ENTER** key without entering any number:

Standard Deviation (So) =
<default: 0.45>
<press ENTER>

If you wish to use a different value, type the value in the input-cell, and press the **ENTER** key:

Standard Deviation (So) = 0.40
<default: 0.45>
<press ENTER>

After you have entered values for resilient modulus, reliability, and standard deviation, you will be requested to enter the

Structural Number of the pavement structure. An input screen is displayed:

Highway ID: (data from Section 3.2)

Structural Number

1. **Estimated based on structural layer coefficients (ai), layer thickness (Di), and drainage coefficients (mi)**
2. **Value from Design Specifications.**

Select a choice:

An input-cell is positioned next to the **Select a choice** message. To select an option, type in the number to the left, and then press the **ENTER** key. Depending on the option chosen:

1. The structural number is computed as described in Section 2.3.5 of the 1986 AASHTO Guide (AASHTO, 1986). An input-screen is displayed on the bottom part of the screen:

	ai	Di	mi
Surface Course			
Base Layer			
Subbase Layer			

An input cell is positioned on the **ai** column of the **Surface Course** row. Enter a value for **ai** and then press the **ENTER** key. The input-cell will move to the **Di** column. After you have entered the remaining data, the structural number screen may remain visible for a few seconds during which the structural number is being computed. For example:

Surface Course:
ai: 0.42 <press ENTER>
Di: 8 <press ENTER>

Base Layer:
ai: 0.14 <press ENTER>
Di: 7 <press ENTER>
mi: 1.2 <press ENTER>

Subbase Layer:
ai: 0.08 <press ENTER>
Di: 11 <press ENTER>
mi: 1.2 <press ENTER>

After you have entered the above data, the structural number is computed. The Performance Measure Menu described in Section 3.5.1.3 will be displayed. If you make a mistake, or wish to switch to option 2, refer to Section 3.5.1.2, Editing Structural Data, for instructions.

2. If you know the value for structural number, select option 2. The message:

   Enter a value for Structural Number ==> SN =

   will be displayed on the bottom part of the screen. Next to it, an input-cell is shown. Type in the value for structural number and then press the **ENTER** key. For example:

   Enter a value for Structural Number ==> SN = 8
   <press ENTER>

   The Performance Measure Menu described in Section 3.5.1.3 will be displayed next. If you wish to switch to option 1, press the F8 key, and refer to Section 3.5.1.2, Editing Structural Data, for further information.

3.5.1.2 Editing Structural Data

If you make a mistake when entering structural data, press the F8 key to call the editing routine. The active keys when editing will be displayed on the upper part of the screen:

↑ ↓ : vertical ← → : lateral ENTER : edit ESC : exit

How these keys work has been described in previous sections of this manual (Sections 3.3.5.1 and 3.3.6.1). Refer to Section 3.5.1.2.2 if you call editing from within the structural number routine, or Section 3.5.1.2.1 if you call it from within the resilient modulus routine.

3.5.1.2.1 Edit Mr, R

When you call the editing routine, the highlighted data-cell is positioned in the **Select a choice** option of the resilient modulus data. If you want to edit:

1. **Resilient Modulus:** To change the data entered before, move the data-cell to the Mr option by using the arrow keys (↑ ↓). Press the **ENTER** key to clear the data-cell. Type in the new value and press **ENTER**. Use the **ESC** key to exit editing and continue to execute PBA.

   To change the option for Mr, press the **ENTER** key when the data-cell is on the **Select a choice** row. This will clear the data-cell and the resilient modulus value. Type in a new option, and press **ENTER**. Refer to the procedure explained in Section 3.5.1.1 to enter data for the selected option.

2. **Reliability:** After calling editing, the highlighted data-cell is positioned on the **Select a choice** row of the resilient modulus data set. Use the arrow keys to move the data-cell down to the **Reliability** value. Press the **ENTER** key, and type in the new value, or press **ENTER** without entering a value if you want to use the default value. Press **ESC** to exit editing and continue entering more data.

3.5.1.2.2 Edit SN

After calling the editing routine, the edit screen is displayed on the top of the screen, and a highlighted data-cell is positioned on the **Select a choice** message. At this point, you can do one of two things:

1. Switch structural number option. To switch between options, press the **ENTER** key, this clears the data-cell. Type in the number of the new option. If it is different than the old option, then the bottom part of the screen will be cleared, and the data-frame for the new option will be displayed. Refer to Section 3.5.1.1 for

instructions on entering data for the new option.

2. Edit layer coefficients, layer thickness, and drainage coefficients. Use the arrow keys to position the data-cell on the data you want to modify. Then, press the **ENTER** key, this clears the data-cell. Type the new value, and press **ENTER.** Use the arrow keys to position the data-cell on a different value that you would like to change. Press the **ESC** key to exit editing and continue to execute PBA.

3.5.1.3 Performance Measure

Up to this point, you have entered the data that can be used to compute the Theoretical PBA Index based on PSI, and part of the data needed if deflection is to be used. To select which default performance measure to use, you are presented with a Performance Measure Menu:

Performance Measure

1. Present Serviceability Index (PSI)
2. Deflection
3. Cancel data for current section
Enter a choice:

To select a measure, type the number to the left in the input-cell and press the **ENTER** key. Refer to the following sections for further information.

3.5.1.3.1 PSI

Equation 1.2.1 of the AASHTO Guide (AASHTO, 1986) was rearranged to yield PSI as a function of ESALs and pavement structural data. This equation is used by PBA to compute the Theoretical PBA Index of a highway section as described in Section 3 of Volume I (Ordonez and Vinson, 1988). To select this option, type 1 in the input-cell:

Enter a choice: 1
<press ENTER>

If you selected deflection as the measure of pavement performance, then nothing will happen; the blinking cursor will be positioned back at the beginning of the input-cell. On the other hand, if you selected PSI, then a wait message will be displayed in the middle of the screen:

The PBA Index is being computed
WAIT

After a few seconds, the PBA Index is displayed on the screen. Refer to Section 3.5.1.3.4 for more information.

3.5.1.3.2 Deflection

If you select deflection to compute the Theoretical PBA Index, then you will be requested to enter data about the NDT equipment and the pavement structure. To select this option, type 2 on the highlighted input-cell and press the **ENTER** key:

Enter a choice: 2
<press ENTER>

On the bottom of the screen, a series of labels is displayed to use when entering the above data:

Deflection Data
Highway ID: (identification number)

Dynamic Load <P, lbs> :
Load Plate Radius <ac, inch> :
Pavement Thickness <h, inch> :
Subgrade's Poisson Ratio <u> :
Field Structural Number <a> :
Design Deflection Value :

F1 : Help   F8 : Edit   ESC : Quit   F9 : End

Information about the data can be found on the AASHTO Guide (AASHTO, 1986). The Field Structural Number is the structural number of the pavement structure on the field at the time the PBA Index is to be computed. Design Deflection Value is the deflection that the pavement structure is assumed to have on the field when constructed, or immediately after an overlay has been placed on top of the pavement structure. For example:

Dynamic Load <P, lbs> :	**4500**	**<press ENTER>**
Load Plate Radius <ac, inch>:	**6**	**<press ENTER>**
Pavement Thickness <h, inch>:	**12**	**<press ENTER>**
Subgrade's Poisson Ratio <u>:	**0.45**	**<press ENTER>**
Field Structural Number :	**5**	**<press ENTER>**
Design Deflection Value :	**0.001**	**<press ENTER>**

If you make a mistake press the **F8** key, this will display an editing window on the upper part of the screen:

↑ ↓ : vertical   ENTER : edit   ESC : exit

The function of these keys has been previously explained (see Section 3.3.5.1). A highlighted data-cell is positioned on the **Dynamic Load** row. Use the arrow keys to move the highlighted data-cell down to the data you would like to modify. Press the

ENTER to clear the data-cell. Type in the corrected value and press **ENTER**. Press the **ESC** key to exit editing and continue executing PBA.

After you have entered the deflection data, the computer will take a few seconds to compute the PBA Index. The message:

The PBA Index is being computed
WAIT

is displayed in the middle of the screen. After a few seconds the PBA Index will be displayed. Refer to Section 3.5.1.3.4 for more information.

3.5.1.3.3 Cancel data for current section

If you select option 3, you will return to the highway identification screen described in Section 3.2. The data entered up to this point will be lost. You can enter new data as described previously.

3.5.1.3.4 Performance Plot

After the PBA Index is computed, it is displayed on the middle of the screen, together with the option to see a plot of the performance measure vs the independent variable.

Highway ID: (identification number)
PBA Index = (PBA value and PBA dimension)
Do you want to plot the (PBA variables) curve (y/n) ?

PBA variables and dimension depend on the performance measure selected, i.e. either PSI-ESAL or Deflection-SN. If you would like to see the plot, type **Y** in the input-cell and press the **ENTER** key. The plot will be shown on the screen. Pressing any key will return you to the main data-screen described in Section 3.2. You can enter data for a new highway section or return to the PBA Main Menu. If you type **N**, you will return to the main data screen.

3.5.2 User Defined Performance Equation

If you do not want to use the AASHTO equations, PBA has an option used to enter your own performance model. Refer to Section 5 of this manual for instructions on how to enter your own performance equation. To select this option, type in a **2** and then press the **ENTER** key.

3.5.3 Cancel Data for current section

If you select option 3 of the performance model menu, you will return to the main data-screen described in Section 3.2. The data entered up to this point will be lost.

4. Field Performance PBA Index

The determination of the true total pavement deterioration is achieved by using field performance curves and a quantification of level of routine maintenance, as described in Section 3 of Volume I (Ordonez and Vinson, 1988).

For the determination of the Field PBA Index, PBA allows the user to enter his/her own performance equation, and the data to be used for the variables in that equation. To compute the Field PBA Index, select option 2 of the main menu (see Section 2):

```
                PBA Main Menu
    Oregon Department of Transportation
              Highway Division
         Performance Based Approach

  1. Compute Theoretical-Performance PBA Index
  2. Compute Field-Performance PBA Index
  3. PBA Index for Zero-Level of Routine Maintenance
  4. Pavement Deterioration Proportions
  5. Exit to DOS
```

Using the arrow keys, position the highlighted menu-cell in option 2 and press the **ENTER** key. This will bring up a second menu shown below:

```
          Performance Based Approach
        PBA Index : Field Performance

  1. Compute Field-Performance PBA Index
     for a pavement section.
  2. Compute PBA-Index for Zero-Level of Routine
     Maintenance for a Highway Section.
  3. Return to PBA Main Menu

     Enter a choice:
```

An input-cell is positioned next to the **Enter a choice:** message. Type 1, and press **ENTER**. This will send you to the User Equation routine of PBA. Refer to Section 5 of this manual for more information on how to create a customized performance equation.

5. User's Performance Equation

Specific to this project, was the creation of a correlation between PSI and ODOT's PCR in order to have an equation for use in predicting performance due to load-related factors. However, given that neither a correlation was established, nor a performance model based on PCR has been developed, PBA was enhanced to allow the use of a customized predicted performance equation. This allows the user to create a performance model for computing either theoretical or field performance PBA Index.

5.1 Equation File Menu

Upon calling this routine, the User's Equation Main menu will be displayed:

```
        Performance Equation - Data File

    1. Create a new Equation File
    2. Use an existing Equation File
    3. View an Equation File
    4. Return to Previous Menu
    Enter a choice:
```

A highlighted input-cell is displayed next to the **Enter a choice:** message. These options are described in the following sections.

5.1.1 New Equation File

Select this option if you want to create a new data file on which to store the variables forming the performance model. Type **1** in the input-cell:

Enter a choice: 1
<press ENTER>

The following instructions are displayed on the screen:

New Performance Equation File
(Section 6.2.1: New Pavement Deterioration Data File)
(Section 7.1: New Pavement Deterioration Proportions File)

Enter the name of the data file to be created
You can add a disk drive address (e.g. A:) or a directory path (e.g. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

An input-cell is displayed below this message. By default the data file will have the extension **.EQN** (Section 6.2.1: **.LRA**; Section 7.1: **.SHR**). If you typed in an extension, it is likely that an error message will be displayed on the bottom of the screen. Type in the name of the file, for example:

C:\pba\test
<press ENTER>

One of three things may happen next:

1. The file name entered does not correspond to an already existing file, then, the Equation Menu described in Section 5.2 will be displayed. (From Section 6.2.1: The input data screen described in Section 6.3 will be displayed). (From Section 7.1: the data screen described in Section 7.5 will be displayed).

2. The file name entered corresponds to an existing file. If this is the case, you are presented with two options displayed in the middle of the screen:

   C:\PBA\TEST already exists.
   Do you want to overwrite/delete it (y/n) ?
   Do you want to add new information (y/n) ?

   An input-cell is positioned next to the **overwrite/delete** option. Then:

a. If you type **Y** the following message is displayed:

The data stored in "file name"
are to be deleted.
Enter 'y' to continue, 'n' to cancel

The input-cell is now next to the word **cancel.** Entering a **Y** will erase the file, and create a new file with the given name. The data previously stored in it will be lost. Entering a **N** will cancel erasing the existing file, and will return you to the Equation File Menu. (From Section 6.2.1: you will return to the Total Pavement Deterioration File Menu). (From Section 7.1: you will return to the Deterioration Proportions File Menu of Section 7).

b. If you type in an **N**, the input-cell moves down next to the second option. Entering **Y** will append the data to be entered to the end of the data file, and the Equation Menu described in Section 5.2 will be displayed. (From Section 6.2.1: The pavement deterioration file menu described in Section 6.1 will be displayed). (From Section 7.1: the data screen described in Section 7.5 will be displayed). Entering an **N** will return you to the file menu, and you will have to enter a new file name as previously described.

3. An error message will be displayed at the bottom of the screen. For further information on the error message, refer to Section 8 of this User's Guide. Press the **ESC** key to return to the file menu.

5.1.2 Existing Equation File

This option opens an existing equation file, and then appends the performance model to the end of this file. To select this option type 2 in the input-cell:

Enter a choice: 2
<press ENTER>

Instructions to enter the file name will be displayed:

Existing Performance Equation File
(Section 6.1.2: Existing Pavement Deterioration File)
(Section 7.2: Existing Pavement Deterioration Proportions File)

Enter the name of the data file to be opened
You can add a disk drive address (e.g. A:) or a directory path (e.g. C\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

A highlighted input-cell is displayed below this message. Type in the file name. For example:

C:\pba\test
<press ENTER>

One of two things may happen:

1. A file with that name is found. Refer to paragraph 2 of Section 5.1.1.

2. If a file with that name is not found, an error message will be displayed at the bottom of your screen. Press the ESC key to return to the file menu and enter a new file name. If a different error message is displayed, refer to Section 8 of this User's Guide.

5.1.3 View an Equation File

This option allows you view the contents of an equation data file. To select this option type 3 in the input-cell:

Enter a choice: 3
<press ENTER>

Instructions to enter the file name will be displayed:

View Performance Equation File

Enter the name of the data file to be seen
You can add a disk drive address (e.g. A:) or a directory path (e.g. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).

One of two things may happen:

1. A file with the given name is found. The data stored in the file will be displayed one record at a time. The information displayed is:

Record No: | **PBA Index:**
Performance Measure: | **Highway ID: (or Pavement)**
Equation ID: | **Performance Design Value:**
Perf. Units: | **X var Units:**

	Var 1	**Var 2**	**Var 3**	**Var 4**	**Var 5**
Var Coe					
Log Var					
10 Var					
Ln Var					
e Var					
(c) Var					
Exp Var					
Root Var					
(C)p(V)					
V Rel					

Var. 1: | **Var. 2:**
Var. 3: | **Var. 4:**
Var. 5: | **No. Var.:**

ENTER : continue **ESC : Quit**

The meaning of these data is defined in following sections of this manual. The values 1000000, 2000000, 3000000, 4000000, and 5000000 are codes used by PBA. A 1000000 code means that the operation is not active for the equation shown. The other codes mean that their respective operations are active.

To see the data for different models, press the **ENTER** key. If you wish to return to the Equation File Menu, press the **ESC** key.

2. If a file with that name is not found, an error message will be displayed at the bottom of the screen. Press the

ESC key to return to the Equation File Menu, and try with a different file name. If a different error message is displayed, refer to Section 8.

5.1.4 Return to Previous Menu

If you wish to return to the PBA Index: Field Performance menu described in Section 4, select option 4.

5.2 Performance Equation Menu

After you have successfully opened, or created, a data file, you will be requested to select one of two options for the performance equation: either use an existing equation, or create a new equation. However, if you selected to create a new equation file, option 2, **Use an existing Performance Equation**, will not be active. The equation menu is shown below:

Performance Equation

1. Create a New Performance Equation
2. Use an existing Performance Equation
3. Return to previous menu

Enter a choice:

A highlighted input-cell is displayed next to the **Enter a choice** message. Each of these options is described below.

1. New Performance Equation: Select this option to create a brand new performance equation. Type **1** in the input-cell and press the **ENTER** key. This option is active for both new or existing equation file options of Section 5.1. A screen to enter the identification data for the equation (see Section 5.3) will be displayed.

2. Existing Performance Equation: If you would like to use an equation that you created on a past session with PBA, select option 2 on the Equation Menu above. This option is active only if you selected the **Use an Existing Equation File** option of Section 5.1. To select this option, type 1 in the input-cell and press the **ENTER** key. This will bring up the screen for entering the identification data of the equation, as described in Section 5.3.

3. Return to Previous Menu: This option lets you return to the menu from where you called the User's Equation Routine.

5.3 Performance Equation, Identification Data

If you want to create, or use, a performance equation you will be requested to enter data for identifying the equation, the pavement section on which the equation is applied, and the performance measure used. The input-screen, shown below, will be displayed to guide you in entering the data needed.

```
     User-Defined Performance Equation
    Enter information for MAIN EQUATION

      Performance Measure        :
      Performance Units          :
      Performance Design Value   :
      Maximum Performance Scale  :
      Minimum Performance Scale  :
      Pavement Section           :
      Equation ID Number         :
      Number of Sub-Equations
                  (2 minimum)    :

      F3 : edit     ESC  : quit
```

A highlighted input-cell is located next to the **Performance Measure** label. A description of each data follows.

1. Performance Measure: Enter the name of the performance measure that will be evaluated with the equation (e.g. PSI, deflection, PCR, etc.). You have up to 13 characters for describing the measure. For example:

   Performance Measure : PSI
   <press ENTER>

2. Performance Units: Enter the dimension of the performance measure (e.g. inch for deflection). You have up to 13 characters. For example:

   Performance Units : PSI
   <press ENTER>

3. Performance-Design Value: PBA determines the pavement deterioration as the cumulative area bound by an initial level of serviceability and a performance curve, as described in Volume I (Ordonez and Vinson, 1988). The initial level of serviceability reflects the condition of the pavement when constructed, or, after major rehabilitation is undertaken. For example, the AASHTO Guide gives an initial PSI value of 4.2 for flexible pavements (AASHTO, 1986). For example:

   Performance-Design Value : 4.2
   <press ENTER>

4. Maximum and Minimum Performance Scale: These data will be used for graphic purposes only. When plotting the Performance vs. X-variable curve, PBA will use these data to generate an appropriate coordinate Y axis to plot the performance values. For example, it is well known that the scale of PSI ranges from 0 through 5:

   Maximum Performance Scale : 5
   <press ENTER>
   Minimum Performance Scale : 0
   <press ENTER>

5. Pavement Section: Enter the identification number of the pavement section (or highway section, when defining equation for Theoretical-PBA Index) on which the equation will be applied. This data will be used later to retrieve the PBA Index for the section. You have up to 13 characters. For example:

   Pavement Section : Test 1
   <press ENTER>

6. Equation ID Number: Enter a string of up to 10 characters to identify the equation, and to use in

identifying the sub-equations that make up the main equation. As you will later see, the equation will be broken down into sub-equations. You can have up to 5 sub-equations, each with as many as 5 variables. To clarify this, assume you have the following performance equation:

PSI = (3.71 X + 0.001 Y - Z) * (Log T)

X, Y, Z, and T are variables used to determine PSI values. To enter this main equation into PBA, we will break it down in two sub-equations:

Sub-equation 1: 3.71 X + 0.001 Y - Z
Sub-equation 2: Log T

Now, sub-equation 1 will have 3 variables:

Variable 1-1 : 3.71 X
Variable 1-2 : 0.001 Y
Variable 1-3 : - Z

and for sub-equation 2:

Variable 2-1 : Log T

To continue with the description of the identification data, and with our example, enter:

Equation ID Number : 1
<press ENTER>

7. Number of Sub-Equations: As described above, each main equation is broken down into sub-equations. You are asked to enter how many sub-equations will be used. A minimum of 2 is required for a proper execution of PBA.

   Number of Sub-Equations
   (2 minimum) : 2
   <press ENTER>

After you have entered the identification data, and, if you are creating a new equation, you will be requested to define the sub-equations forming the main equation. Refer to Section 5.4 for further instructions. If you are retrieving an existing equation, you will be asked to enter data for the different variables in the equation. Refer to Section 5.5 for more information.

There are two active keys displayed at the bottom of the data frame. If you would like to edit any data entered, press the F3 key. This will clear the data-cells and position the highlighted input-cell on the **Performance Measure** label. If you would like to terminate the process of creating or retrieving an equation, press the ESC key, this will send you back to the Performance Equation Menu described in Section 5.2 of this User's Guide.

5.4 New Performance Equation

The next step on creating a performance equation is to enter the variables which form the sub-equations. A screen showing different options for defining a variable will be displayed:

```
          User's Performance Equation

     Performance Measure   :
     Pavement Section #    :
     Equation ID number    :
     Variable ID number    :

     Variable Description :
     Equation Elements :
       Coefficient            :
       Log (variable)         :
       10^(variable)          :
       Ln (variable)          :
       e^(variable)           :
       variable*(constant)    :
       Constant^(variable)    :
       (Variable)^(exp)       :
       (Variable)+constant    :
       Variable Relation      :

       F1: Help    F2 : Edit    ESC : Quit    ↑ ↓ : Up, Down
```

A highlighted input-cell will be displayed next to the **Variable Description** label. On the upper part of the screen, the description of the performance measure and the pavement (or highway) section ID will be displayed. The string, displayed in the **Equation ID number** cell, will be formed by two parts. The characters to the left of the dash are the characters you enter to identify the main equation in Section 5.3. The number to the right of the dash is the number of the sub-equation you are currently creating. The number in the **Variable ID number** cell, is the number of the sub-equation's variable you are entering information on. The other data are described below.

1. **Variable Description:** Enter a description of the variable; you can enter up to 24 characters. There is a special word that PBA uses to identify variables formed by a numeric constant. This word is **CONSTANT**. If, for example, you have the following sub-equation:

   Sub-equation : 3.91 - 0.0071 R

   you can break it down into two variables:

   Variable 1: 3.91
   Variable 2: - 0.0071 R

   where 3.91 is a numeric constant. Therefore, when entering this number, type in the word **Constant** (either uppercase or lowercase). This will save some time by skipping the remaining options and moving on to the next variable. For example:

   Variable Description : constant
   <press ENTER>

   or

   Variable Description : Roadmeter counts
   <press ENTER>

   The input-cell moves down to the **Coefficient** cell.

2. **Coefficient:** A numeric constant, or number multiplying a variable is defined as coefficient by PBA. For example:

    Coefficient : 3.91
    <press ENTER>

    or

    Coefficient : -0.0071
    <press ENTER>

    or

    Coefficient :
    <press ENTER>

    If you typed in a constant, refer to Section 5.4.2 for further information on creating the performance model. If you entered any other variable, keep reading this section.

3. **Log (variable):** After you enter the coefficient for the variable, the blinking cursor moves down to the **Log (variable)** cell. Use this option if you want to compute the logarithm of base 10 of the variable. For example:

    Sub-equation: Log (3 * R)

    To evaluate the logarithm of (3 * R) select this option. If you press the **ENTER** key to select this option, a lower case **x** will be displayed on the option-cell and the blinking cursor will move down to the **Variable*(constant)** option. If you don't want to select this option, use the arrow keys (↑ ↓) to move the cursor down through the next three options.

4. **10^(variable):** this option will evaluate 10 to the "variable" power. For example:

    Sub-equation: 3 + 10^(deflection)

    This option will evaluate 10 to the "deflection" power, where deflection is a variable (to be substituted by numeric data as described in Section 5.4.1.) To select this option, press the **ENTER** key when the blinking cursor is positioned in the option-cell. If you don't want to use this option, use the arrow keys (↑ ↓) to move the cursor down to the next option.

5. **Ln (variable):** This option evaluates the natural (base e) logarithm of the variable. For example:

Sub-equation: 3.31 + ln(3.92R)

Enter 3.31 as for code 1, then use the arrow keys to move the cursor down to option 5, and press the **ENTER** key. Next enter 3.92 as in option 7. If you don't want to use this option, use the arrow keys to move the cursor to the next function.

6. **e^(variable):** This option evaluates "e" to the "variable" power, where "variable" is a numeric constant (see Section 5.4.1), and "e" is the base for natural logarithms (approximately 2.718282). For example:

   Sub-equation: 3.76 + e^(roadmeter counts)

   Position the cursor in the option-cell of this option, and press the **ENTER** key. A lower case **x** will be displayed in the option-cell, and the cursor will move down to the **Variable*(constant)** option. If you don't want to use this option, use the arrow keys to position the cursor on the next function.

7. **Variable*(constant):** This option allows you to enter a numeric constant which multiplies the variable. This option works as the **Coefficient** option above, if the constant is multiplying the variable itself. For example:

   3.91 * R

   then you can use either option to enter the 3.91. However, if you have a different structure, one where the variable is affected by other options before it is multiplied by the constant, then use option 7:

   3.91 (Log R)

   then,

   Variable*(constant) : 3.91
   <press ENTER>

   the blinking cursor will move down to the **constant^ (variable)** option. If you don't want to use this option, press the **ENTER** key, when the blinking cursor is on the option-cell, without entering any data. The cursor will move down to the next option.

8. **Constant^(variable):** This option evaluates "constant" to the "variable" power, where "variable" is numeric data (see Section 5.4.1), and "constant" is a numeric constant entered using this option. For example:

Sub-equation: 1.5^(R + 3.9)

Enter 1.5 using this option, and 3.9 as in option 10.

Constant^(variable) : 1.5
<press ENTER>

The cursor will move down to the next option. If you don't want to use this option, press the **ENTER** key when the cursor is on the option-cell without entering any data.

9. **(variable)^(exp):** This option evaluates "variable" to the "exp" power, where "variable" is numeric data (see Section 5.4.1), and "exp" is a numeric constant entered using this option. For example:

   Sub-equation: R^(2)

   (variable)^(exp) : 2
   <press ENTER>

   The blinking cursor will be positioned in the option-cell of the following option. If you don't want to use this option, press the **ENTER** key when the cursor is on the option-cell without entering any data. The cursor will move down to the next option.

10. **(variable) + constant:** This option adds "constant" to "variable", where "variable" is a numeric value (see Section 5.4.1), and "constant" is a numeric constant entered using this option. For example:

    Sub-equation: 1.5^(R + 3.9)

    (variable) + constant : 3.9
    <press ENTER>

    If you don't want to use this option, press the **ENTER** key when the cursor is in the option-cell without entering any data.

There is an order of operations followed by PBA. The first option performed when evaluating the equation, is **(variable) constant**, and the last option performed is **Coefficient**. In other words, the order of precedence is:

Order	Option
1 st	(variable)+constant
2 nd	(variable)^(exp)
3 rd	Constant^(variable)
4 th	Variable * constant
5 th	e^(variable) or, Ln (variable) or, 10^(variable) or, Log (variable)
6 th	Coefficient

For example, assuming you selected all of the options, PBA will compute first **(variable)+constant.** The result will be substituted for **variable** in the **(variable)^(exp)** option. This last result would be used by the third option **Constant^(variable)** as input for **variable**, and so on.

11. **Variable Relation:** The last option on the screen is used to relate the result obtained from a variable, to the result from the other variables; or to perform other functions, which were not easily defined with the above options. For example:

    Main equation: **(3X + 2Y) * (Z^(2))**

    Sub-equation 1: **3X + 2Y**
    Sub-equation 2: **Z^(2)**

    This option will allow you to multiply the result obtained from (3X + 2Y) times the result from (Z^(2)).

    The variable relations are entered as codes, the first part of which are listed below:

Code	Meaning
0	The result of the current variable is added to the partial result of the previous variables.
10	Evaluates the logarithm of base 10 of the partial result of the previous variables.

20	Evaluates 10 to the "partial result" power.
30	Evaluates the natural (base e) logarithm of the partial result.
40	Evaluates "e" to the "partial result" power, where "e" is the base for natural logarithms, approximately 2.718282.
50	The "variable result" will be multiplied times the partial result of the previous variables.
60	The partial result of the previous variables is divided by the result of the current variable.
70	The result of the current variable is added to the partial result of the previous variables.
80	The result of the current variable is substracted from the partial result of the previous variables.
90	The partial result of the previous variables is multiplied times a numeric constant, entered in option 2 (Coefficient).

Partial result of previous variables refers to the result obtained from combining, through mathematical functions, the result of each variable with the result of the variables entered before it. For example:

Sub-equation: Log (3X + 5Y - Log Z)

If the current variable is (- Log Z), then the partial result of previous variables refers to the result obtained from evaluating (3X + 5Y). The result of the current variable will be the value yield by (- Log Z). When entering the data for the variables, enter in the Variable Relation cell:

Variable 3X:

Variable Relation : 0

Variable 5Y:

Variable Relation : 70
(this will add the result from 3X to the result of 5Y)

Variable (-Log Z):

Variable Relation : 80
(this will substract the result of
-Log Z from the result of 3X+5Y)

To complete this example, and to compute the logarithm of base 10 of the partial result (3X + 5Y - Log Z), use Code 10. To do this, create a variable for which you do not enter any data but a 10 on the Variable Relation option, i.e.:

Variable Description	:		<press ENTER>
Coefficient	:		<press ENTER>
Log (variable)	:		<press arrow key>
10^(variable)	:		<press arrow key>
Ln (variable)	:		<press arrow key>
e^(variable)	:		<press arrow key>
Variable * constant	:		<press ENTER>
Constant^(variable)	:		<press ENTER>
(variable)^(exp)	:		<press ENTER>
(variable)+constant	:		<press ENTER>
Variable Relation	:	10	<press ENTER>

The second part of the codes for the variable relation are used to account for operations with numeric constants and other operations that couldn't be handled with options 2 through 10.

Code	Meaning
1XX.YYYYYYY	This option will evaluate the partial result of previous variables to the (1/YY.YYYYY) power. Use 1XX.YYYYYYY where "YYYYYYY" represents a real number of up to 7 figures. For example:

(Result)^(1/2)

Variable Relation : 110.2
<press ENTER>

The first X to the rigth of the number 1, represents the number of integer digits on the number. The X to the left of the decimal point represents the number of figures in the fractional part of the real number. The Xs to the right represent the figures which form the real number. For example, the number 2.7182 would be represented as:

114.27182

the number 243.16 as:

132.24316

2XX.YYYYYYY — This option will evaluate the partial result of previous variables to the (YY.YYYYY) power. The meaning of the Xs and Ys is given in code 1.

3 — This code will yield the reciprocal of the partial result of previous variables. The only element for the variable will be this code. For example:

Sub-equation: 1/(3X + 2Z)
Variable 1: 3X
Variable 2: 2Z
Variable 3: 3 (Variable Relation)
Partial result: (3X + 2Z)

This code would yield:

1/(partial result)

4XX.YYYYYYY — This option will add the value represented by the Ys to the result of the current variable, and then add this last value to the result of the previous variables. This option is convenient when you don't want to define a single numeric constant as a sub-equation. For example:

Sub-equation: 1.25+(3X+2Z+Y-(T/R))

There are 5 variables in the above sub-equation. It would be inconvenient to define a sub-equation with a unique value of 1.25. Therefore, after defining the first four variables, enter for the last variable:

Variable Relation: 42.125
<press ENTER>

This will add the result of 3X to 1.25 and then this to the result from

(2Z+Y(T/R)).

5XX.YYYYYYY — This option will substract the value represented by the Ys to the result of the current variable, and then add this last result to the partial result of previous variables. See example above, but use -1.25 instead.

6XX.YYYYYYY — This option will add the numeric constant represented by the Xs to the partial result of the previous variables. The difference between this code and code 4 is that code 6 is defined as a unique variable, while code 4 is defined as part of the last variable. Use code 6 when it is not possible for you to define the sub-equation using code 4.

7XX.YYYYYYY — This option will substract the numeric constant represented by the Xs, from the partial result of the previous variables. The difference between this code and code 5 is that code 7 is defined as a unique variable, while code 5 is defined as part of the last variable. Use code 7 when it is not possible for you to define the sub-equation with code 5.

A special type of sub-equation is the **Dummy** sub-equation. Its applications will be explained in Sections 5.5 and 9. To create this sub-equation enter the following:

Variable Description : **dummy**
<press ENTER>

And fill in the different options needed. The dummy sub-equation will be helpful when defining the relationships between sub-equations. For example:

Main Equation : Log[(3X + 5Y) * (2Z + T)]

The main equation can be broken down into:

Sub-equation 1: 3X + 5Y
Sub-equation 2: 2Z + T
Dummy : 1

When relating the sub-equations (see Section 5.5):

A = 1-2
B = A-dummy

Relation B would be defined as the Log of the result from relationship A.

After you have defined a sub-equation, the following active keys will be displayed on the bottom of the screen:

1. If you defined "variable" as a constant:

   Press ENTER to CONTINUE entering variables
   E to END creation of sub-equation

   If you press the **ENTER** key, the data frame will be cleared, and you will be requested to enter information for a new sub-equation. If you press the **E** key, you will terminate the creation of the sub-equation, and you will be asked to enter the variables for the remaining sub-equations; or, if you are in the last sub-equation, PBA will request that you define the relationship between the different sub-equations (see Section 5.5).

2. If you defined the variable as something else:

   Press ENTER to INPUT data for the variable
   E to END creation of sub-equation
   C to enter more variables

   Data to be substituted for "variable" in the previous codes and options, can be input if you press the **ENTER** key. This will call the data input screen described in

Section 5.4.2. If you press the **E** key, you will terminate the creation of the sub-equation, and you will be asked to enter the variables for the remaining sub-equations; or, if you are in the last sub-equation, PBA will request that you define the relationship between the different sub-equations (see Section 5.5). If you press the **C** key, you can enter data for a new variable as described in the above paragraphs.

5.4.1 Edit Variable Coefficients

If you make a mistake when defining a variable, press the **F2** key. This will clear the input-screen, and the cursor will be positioned in the **Variable Description** label. You can enter the correct data for the variable. With this version of PBA, you can not edit variables or sub-equations that you created before.

5.4.2 Variable Data Input

After you create, or retrieve, a variable, you will enter the numeric data to be substituted for the variable in the main equation. The data-input screen will be displayed:

```
Pavement ID  :    (ID number)
Equation ID  :    (ID number)
Variable     :    (variable description)
No. Variable :    (variable number)
Number of measurements (max: 10) :

-------------------------------------
| Values for variable (variable ID) |
|  YEAR               MEASURE       |
-------------------------------------
|                                   |
|                                   |
|                                   |
-------------------------------------

F1 : Help    F4 : Edit    ESC : Quit    F9 : End
```

A highlighted input-cell is displayed next to the **(max: 10)** message. Enter the number of data (measurements) to be used. A number between 1 and 10 is expected. (Any other number will make the message blink and clear the input-cell.) After you enter the number of data to input, and equal number of input-cells will be highlighted on the data frame on the bottom of the screen, and the blinking cursor will be positioned on the input-cell for the YEAR column on the first row. To input the data, type in the values for each column and row, and then press the **ENTER** key. It is not necessary to input a value for the **YEAR** column; it is part of PBA to help guide you on inputing the data. For example, for a variable representing roadmeter counts:

Number of measurements (max: 10) : 5
<press ENTER>

Five input-cells will be displayed for each column on the data frame, and the cursor is positioned in the **YEAR** column for value number one. Type in the following values:

1975
<press ENTER>

the cursor moves to the **MEASUREMENT** column, enter:

1000
<press ENTER>

the cursor moves down to the **YEAR** column of value No. 2, enter:

1977 <press ENTER>
1200 <press ENTER>
1980 <press ENTER>
1400 <press ENTER>
1984 <press ENTER>
1700 <press ENTER>
1986 <press ENTER>
2000 <press ENTER>

If you need to edit a value, refer to Section 5.4.1.1 for instructions on how to call the Editing Routine.

After you have entered the data for the variable, you will return to the variable input-screen described in Section 5.4. Two options are displayed at the bottom of the screen:

Press ENTER to CONTINUE entering variables
Press E to END creation of equation

If you press the **ENTER** key, you can enter data for a new variable as described in Section 5.4. If you press the **E** key, you will terminate the creation of the sub-equation, and you will be either asked to enter the variables for the remaining sub-equations, or, if you are in the last sub-equation, PBA will request that you define the relationship between the different sub-equations (see Section 5.5).

5.4.2.1 Editing Variable Data Input

To edit mistakes, when entering the values for the variables, press the **F4** key to call the editing routine. This will display an editing screen on the upper part of the input screen:

↑ ↓ : vertical   ← →: lateral   ENTER : edit   ESC : exit

The data-cell on the **Number of measurements** message will be highlighted. A description of these keys is given below:

↑ ↓ : vertical	Arrow keys are used to move the highlighted cell in a vertical direction. The data-cell will move downward every time one of these keys is pressed. When the data-cell reaches the last position of the input-cell before editing was called, the data-cell moves to the first row. Use these keys to position the highlighted cell where you want to edit data.
← →: lateral	These arrow keys are used to move the highlighted cell in a lateral direction. The data-cell will move to the column to the right every time one of these keys is pressed. When the data-cell is on the **MEASURE** column, the data-cell moves to the **YEAR** column. Use these keys to position the data-cell where you want to edit data.
ENTER : edit	When the highlighted data-cell is located on the data that you want to edit, press the **ENTER** key to input new data. A blank data-cell is displayed and a blinking cursor is positioned on it. Type in the new data, and then press the **ENTER** key to finish editing. The data-cell shows the new data. Use the arrow keys to move the data-cell to a new position.
ESC : exit	After you have edited the mistakes, press the **ESC** key to continue executing PBA. The blinking cursor will be positioned back where it was before editing was called.

If you want to increase the number of measurements, press the **ENTER** key when the data-cell is on the **Number of measurements** option. This will clear the data-cell. Type in the new number of values, and then press the **ENTER** key. If the new value is greater than the old number, then more input-cells will be displayed in the data frame for each column. If the new value is equal or less than the old one, nothing will happen.

If you want to decrease the number of values, just enter a value of zero for each data beyond the number of measurements you want to use. For example, if you entered 7 for the number of measurements, but you only wanted to enter data for 4, then enter a zero for the data of values 5, 6, & 7. You can also press ENTER without any input, when the cursor is positioned in data-cells 5, 6 & 7.

5.4.3 Keys

The active keys during the creation of the sub-equations are:

F1 : Help	This will display a help screen on the middle part of the screen. Parts of this document or of Volume I, will be displayed for quick reference.
F2 : Edit **F4 : Edit**	Calls the editing routines described in Sections 5.4.1 and 5.4.2.1.
ESC : Quit	This key terminates the process of creating a performance equation. You will return to the Performance Equation menu.
F9 : End	This key terminates the execution of PBA. The data entered up to this point will be lost.

↑ ↓ : Up, Down — Use these keys to move the cursor up or down when defining a variable.

5.5 Relationship Between Sub-Equations

The first step on creating a performance equation was to define the sub-equations which form the main equation. The next step is to define the relationship between the sub-equations, in order to obtain the final result. A screen showing the different options available for correlating the sub-equations will be displayed:

```
        Relationship between Performance Sub-Equations
           Highway ID   :
           Equation ID  :
           Performance  :

                       Sub-Equations
                 A      B       C      D

           +
           -
           *
           /
           Exp
           e^
           Ln
           Log
           10^

           A: 1-2   B: A-3   C: B-4   D: C-5
↑↓ : vertical  ENTER : relation  F1 : Help  F2 : Edit  ESC : exit
```

A series of highlighted columns will be displayed under the Sub-Equations heading. Depending on how many sub-equations there are, you will see all of the columns, or just those needed to define the main equation. The cursor will be positioned on the A column of the + row.

The meaning of the columns **A**, **B**, **C**, and **D** is given below. Column **A** defines the relationship between sub-equations 1 and 2. The result will be related to the result from sub-equation 3 through relationship **B**. The result of **A** and sub-equation 3, **B**, will be related to the result of sub-equation 4 by relationship **C**. This last result, **C**, is related to the result of sub-equation 5 through relationship **D**.

The available arithmetic operations are described below, in which Result I refers to the result represented to the left of the dash, and Result J refers to the result to the right of the dash. For example, for relationship B, Result I is the result from relationship **A**, and Result J is the result from sub-equation 3.

Code	Meaning
+	This relation adds Result I to Result J: I + J.
-	This relation substracts Result J from Result I: I - J.
*	This relation multiplies Result I times Result J: I * J.
/	This relation divides Result I by Result J: I/J.
Exp	This relation evaluates Result I to the Result J power: (I)^(J).
e	This relation evaluates "e" to the "Result I" power, where "e" is the base for natural logarithms, approximately 2.718282. In this case, Result J is a dummy sub-equation created by defining its unique variable as a **CONSTANT**, and giving its **Coefficient** a value of 1. This dummy sub-equation must be defined,

after the sub-equation used to obtain Result I is defined.

Ln	This relation evaluates the natural (base e) logarithm of Result I. See code **e** for more on dummy sub-equation.
Log	This relation evaluates the logarithm of base 10 of Result I. See code **e** for more on dummy sub-equations.
10^	This relation evaluates 10 to the Result I power. See code **e** for more on the dummy sub-equation.

To define the mathematic operation for any of the relationships, A, B, C, or D, use the arrow keys (↑ ↓) to position the highlighted cell on the desired operation, then press the **ENTER** key. A lower case **x** will be shown on the operation selected. The operation-cell will move to the + row of the relationshipnext to be defined next. If you want to change a relationship, press the **F2** key. This will clear the relationships defined, and place the operation-cell on the + row of the **A** relationship.

After you have defined the relationship between the sub-equations, PBA will compute the PBA Index. Refer to Section 5.7 for more on the PBA Index, and on the plot of Performance vs. Independent Variable. However, if you are creating a performance equation to compute the Field PBA Index, you will be requested to input the data for Independent variable of the Performance Measure. For example, if you are using PSI, you will enter the data for Traffic, so that the PBA Index can be computed and the PSI-Traffic graph plotted. Refer to Section 5.6 if this is the case.

5.6 Field Performance : Independent Performance Variable, Manual Input

To evaluate the PBA Index, the change in performance should be related to the load related factors (refer to Volume I, Ordonez and Vinson, 1988). After you have evaluated performance by using a customized performance equation, you should enter those data related to a change in performance, in order to compute the Field PBA Index.

The following data input-screen will be displayed:

Field Measurements
Independent Variable of Field Performance Equation

Enter the variable name and units:
Pavement Section ID :
Variable Name :
Variable Units :
Variable ID :

The active keys will be displayed on the bottom of the screen. A highlighted input-cell will be placed next to the **Pavement Section ID** label. Enter an identification string (up to 10 characters) for the section in which the performance equation is being used. For example:

Pavement Section ID : Testing
<press ENTER>

The input-cell moves down to the **Variable Name** label. Type in the name of the variable to be correlated with a change in performance (up to 25 characters). For example:

Variable Name : Traffic
<press ENTER>

Next, enter the units of the above variable. (No more than 25 characters.) For example:

Variable Units : ESAL
<press ENTER>

Now, type in a string of characters which will be used as a heading on the data-input frame, described in following paragraphs. For example:

Variable ID: ESAL
<press ENTER>

After you enter this information, the middle of the screen will clear, and a data frame will be displayed. You will be asked to enter the number of data to be substituted for in the variable:

Input the number of measurements
(maximum of 10 measurements) :

A highlighted cell is positioned next to the **maximum** message. The procedure to enter, and edit data, will be the same as when entering and editing data for the variables used to define the sub-equations of the performance model. Therefore, refer to Section 5.4.2 for more information on how to enter and edit data. After entering the data, the Field PBA Index will be computed. Refer to Section 5.7 for more on this.

5.7 PBA Index, Plot of Performance Curve

After the PBA Index has been computed, it is displayed on a window in the center of the screen. The option to view a plot of the performance curve is also given:

Pavement ID :
PBA Index = (PBA Index Value, PBA dimension)
Do you want to plot the (PBA variables) curve (y/n) ?

If you would like to view a plot of the performance measure vs. its independent variable, type **Y** and press the **ENTER** key. The plot will be displayed on the screen. Pressing any key will return you to the Performance Equation menu described in Section 5.1. If you elect not to see the plot, type **N**, and press the **ENTER** key. You will return to the Performance Equation menu described in Section 5.1.

5.8 Existing Performance Equation

If you want to use a performance equation that you created on a past session with PBA, select option 2 of the Performance Equation menu described in Section 5.2. After you select this option, the identification data screen, described in Section 5.3, will be displayed. Enter the information requested, as explained in that section. After you have entered the number of sub-equations, the performance equation will be retrieved from the equation file. The screen used to input the identification data will remain displayed for a few seconds, after which one of two things may happen:

1. If the equation ID number you entered does not correspond to an equation saved before, a message will be displayed below the ID input-screen:

   The Equation (equation ID) does not exist
   Press any key to continue

   Pressing any key will return you to the Performance Equation menu of Section 5.3.

2. An equation given that identification number is found. The different sub-equations forming the main equation will be retrieved. If you have to enter data for any variable, PBA will display a data-input screen as described in Section 5.4.2. This process may be repeated several times, depending on the number of variables in the main equation.

   After data for the different variables have been entered, you will have to define the relationships between the different sub-equations as explained in Section 5.5. If you are retrieving an equation for computing the Field-PBA Index, you will be requested to input the data for the independent variable of the performance measure, as explained in Section 5.6.

   After entering the data, the PBA Index will be computed and displayed in the middle of the screen. Refer to Section 5.7 for more information on the PBA Index, and on how to view a plot of the performance curve.

6. PBA Index for Zero-Level of Routine Maintenance

As described in Volume I of this report (Ordonez and Vinson, 1988), pavement distress is the result of load and non-load related factors, the interaction of the two, and the "reduction" due to routine maintenance. When routine maintenance is carried out on the pavement sections, some of the deterioration is repaired, which means that the performance measured in the field will be less than the true performance. To account for this, a quantification of level of routine maintenance is to be related to pavement deterioration, as evaluated by the PBA Index. From this, a measure of the true total pavement deterioration of the highway section for a zero level of routine maintenance can be evaluated.

6.1 Zero-Level of Routine Maintenance Menu

After you select option 3 of the PBA Main Menu (see Section 2), a second menu will be displayed:

```
              Performance Based Approach
            PBA Index : Field Performance

    1. Compute Field-Performance PBA-Index
       for a Pavement Section.
    2. Compute PBA-Index for Zero-Level of Routine
       Maintenance for a Highway Section.
    3. Return to PBA Main Menu

       Enter a choice:
```

A highlighted input-cell is positioned next to the **Enter a choice** message. Type in a 2, and press the **ENTER** key. The User

Equation Menu described in Section 5.1 will be displayed.

6.2 Zero-Level of Routine Maintenance Data Files

The first data requested by PBA to compute the Zero-PBA Index is the Field-PBA Index of each pavement section of the highway section. The User Equation Menu displayed allows you to call upon the data file which stores these Field-PBA Indexes:

Performance Equation - Data File

1. Create a New Equation File
2. Use an Existing Equation File
3. View an Equation File
4. Return to Previous Menu
Enter a choice:

However, for this part of PBA, only options 2, 3, and 4 are active. You can not use option 1, create a new equation file, when computing the Zero-PBA Index. To retrieve a data file for use in this section, refer to Sections 5.1.2 through 5.1.4.

After you have called the data file to retrieve the Field-PBA Index values, you will be asked to create, or retrieve, a file to store the Zero-PBA Index. The file menu will be displayed:

Total Pavement Deterioration - Data File

1. Create a New Deterioration data file
2. Use an existing Deterioration data file
3. View a Deterioration data file
4. Return to previous menu
Enter a choice:

A highlighted input-cell is displayed next to the **Enter a choice** message. The above options are described below.

6.2.1 New Deterioration Data File

The creation of a new data file has been explained in Section 5.1.1. Refer to that section, and follow the instructions in parenthesis for Section 6.1.1 when required.

6.2.2 Existing Deterioration Data File

The use of an existing data file has been explained in Section 5.1.2. Refer to that Section, and follow the instructions in parenthesis for Section 6.1.2 when required.

6.2.3 View a Deterioration Data File

This option lets you view the contents of a pavement deterioration data file. To select this option type 3, and press the **ENTER** key:

Enter a choice : 3
<press ENTER>

Instructions to enter the file name will be displayed:

View a Pavement Deterioration File

Enter the Name of the data file to be seen
You can add a disk drive address (e.g. A:) or a
directory path (e.g. C:\PBA\ ). DO NOT ADD A FILE
EXTENSION. (40 characters maximum).

A highlighted input-cell is displayed below this message. By default the file will have the extension .LRA. If you type in an extension, you will see an error message displayed on the bottom of your screen. Enter the file name. For example:

C:\pba\test
<press ENTER>

One of two things may happen:

1. A file with the given name is found. The data stored in the file will be displayed, one record at a time. The information displayed will be:

File Name

Record No.:	**Highway ID:**
Performance:	**Perf. Units:**
Maintenance Level:	**Units:**
Zero PBA Index:	**Units:**
Reg. Eqn. Slope:	**Correlation Coefficient:**

No.	**Pavement Section ID**	**PBA Index**	**Level of Routine Maintenance**

ENTER : continue **ESC : quit**

The different pavement sections, with their respective PBA Index and Level of Routine Maintenance, will be displayed. After you have viewed all of these records, or have exited by pressing the ESC key, you will return to the File Menu described in Section 6.1.

2. If a file with that name is not found, an error message will be displayed on the bottom of the screen. Press the ESC key to return to the file menu and try with a different file name. If a different error message is displayed, refer to Section 8 for further information.

6.2.4 Return to previous menu

If you wish to return to the menu described in Section 6.1, select option 4 of the Total Pavement Deterioration - Data File menu.

6.3 Zero-PBA Index : Input Data

After you have opened a file to retrieve Field-PBA Indexes and to store the Zero-PBA Index, you will be requested to enter the identification number of the pavement sections which form the highway section. A data input-screen will be displayed:

```
                 PBA Index
    Zero Level of Routine Maintenance
    Enter the Highway Section ID:
```

An input-cell is displayed on the Section ID row. Type in the identification number of the highway section (no more than 10 characters) and then press the **ENTER** key. The screen to input the identification data of the pavement section will be displayed next.

```
              PBA Index
     Zero Level of Routine Maintenance
     Highway ID :  (identification number)

     Input the number of Pavement Sections
     (minimum: 2    maximum: 20 sections) :

  -------------------------------------------
  |      Pavement Sections ID               |
  -------------------------------------------
  |                                         |
  |                                         |
  -------------------------------------------
```

A highlighted input-cell will be displayed next to the number of sections message. Type in the number of pavement sections you will use to compute the Zero-PBA Index for the Highway Section. For example:

(minimum: 2 maximum: 20 sections) : 5
<press ENTER>

An equal number of input-cells will be displayed in the data frame. A minimum of two pavement sections is required for the regression analysis. An error message will be displayed if you enter less than two or more than 20 sections. The cursor moves down next to the number one. Type in the identification number and then press the **ENTER** key. (If you make a mistake, refer to Section 5.4.2.1 for instructions on how to edit data.) For example, enter the following data:

Test 1
<press ENTER>

The input-cell moves next to the number two:

Test 2
<press ENTER>
Test 3
<press ENTER>
Test 4
<press ENTER>

The Field-PBA Index for these sections will be retrieved from the file called in Section 6.2. A message is displayed on the bottom part of the screen:

Retrieving Pavement Sections
WAIT

One of two things may happen next:

1. The data for a pavement section were not found. A message is displayed indicating you that a record for that section was not found in the data file:

   Section: (ID number) could not be found

   ENTER : edit   D : delete   ESC : quit

   The active keys are shown below the message. If you wish to enter a different ID number, press the **ENTER** key. An input-cell is displayed:

   Enter new ID:

   Type in the new ID number. The new section will be retrieved from the data file.

   If you press the **D** key, the section will be deleted from the set of pavement sections entered above. For example, if section "Test 4" was not found, it will be deleted, and PBA will use the data for sections Test 1, Test 2, and Test 3 only. If you press the **ESC** key, the process of computing the Zero-PBA Index will be terminated, and you will return to the menu described in Section 6.1.

2. If the data for the pavement sections are found, you will be requested to enter a measure of level of routine maintenance for each. The input-data screen will be displayed:

   PBA Index
   Zero Level of Routine Maintenance
   Enter the measure name and units:
   Measure Name:
   Measure Units:

   An input-cell is displayed next to the **Measure Name** option. Type in a description of the measure and its units (no more than 13 characters long). For example:

   Measure Name : Cost
   <press ENTER>
   Measure Units: dollars/mile
   <press ENTER>

A data- input frame will be displayed on the bottom of the screen:

```
                    PBA Index
    Zero Level of Routine Maintenance
      Highway ID:  (identification number)
      Measure Name:
      Measure Units:

    -------------------------------------------------
    |      Measure of Level of Routine Maintenance     |
    |   Section      Measure       Section    Measure  |
    -------------------------------------------------
    |                                                  |
    |                                                  |
    |                                                  |
    -------------------------------------------------

    F1 : Help    F4 : Edit    ESC : Quit    F9 : End
```

The pavement sections that you entered before, will be displayed on the Section columns of the data frame. A highlighted input-cell is displayed next to the first pavement section in the Measure column. Type in the value of measure for each pavement section. For example:

```
Test 1:        15000      <press ENTER>
```

The cursor moves to the measure column of the second pavement section. Enter the remaining values:

```
Test 2:        12550      <press ENTER>
Test 3:        25400      <press ENTER>
Test 4:        19560      <press ENTER>
```

If you need to edit data, refer to Section 5.4.2.1. The computer might remain in this screen for a few seconds while the regression analysis of these data is being performed. The results will be displayed on the screen:

(highway identification)

Coefficient of Correlation:
PBA Index Intercept:
Regression Line Slope:

No.	Pavement Section ID	PBA Index	Level of Routine Maintenance	Calculated PBA Index

ENTER : continue **ESC : quit**

The PBA Index for a Zero Level of Routine Maintenance is the value displayed on the **PBA Index Intercept** cell. How good a correlation between Field-PBA Index and Level of Routine Maintenance there is, is given by the coefficient of correlation value. Also displayed are the Field-PBA Index for each pavement section, computed as described in Section 4; and, the calculated PBA Index which is computed from the regression equation. If you enter more than 10 pavement sections, press the **ENTER** key to display the data for the remaining sections. Pressing the **ENTER** key again will save these results in the data file. A message will be displayed on the middle of the screen:

The regression data are being saved
WAIT

After a few moments, the regression line will be displayed on the screen. The curve represents the relation between Field-PBA Index and Level of Routine Maintenance. The original data will be depicted by the bright dots on the graphics screen. Other information shown will include the Zero-PBA Index, the correlation coefficient, and the Highway Identification. If you press any key, you will return to the Field PBA Index Menu, described in Section 6.

7. Pavement Deterioration Responsibilities

The determination of pavement deterioration responsibilities is based on the PBA Index values computed in previous sections (see also Section 3.5.4 of Volume I).

To compute the pavement deterioration responsibilities, select option 4 of the PBA Main Menu, described in Section 2. You will be asked to enter the name of the data file which stores the Theoretical PBA Index for the highway section. This index value is computed as described in Section 3.

```
        Theoretical PBA Index
Enter name of data file to retrieve data
You can add a disk drive address (e.g. A:\ ), or a
directory path (e.g. C:\PBA\ ).  DO NOT ADD A FILE
EXTENSION.  (40 characters maximum).
```

A highlighted input-cell is displayed below this message. Type in the name of the data file, and press the **ENTER** key. If the file does not exist, an error message will be displayed on the bottom of your screen. Press the **ESC** key to try with another file name. If the file exists, you will be requested to enter the name of the file which stores the PBA Index for a Zero Level of Routine Maintenance, created in Section 6:

```
   PBA Index-Zero Level of Routine Maintenance

Enter the name of deterioration file to retrieve data
You can add a disk address (e.g. A:\), or a
directory path (e.g. C:\PBA\ ).  DO NOT ADD A FILE
EXTENSION.  (40 characters maximum).
```

Type in the name of the file in the input-cell. If the file does not exist, an error message will be displayed on the bottom of the screen. Press the **ESC** key and try with a different file name.

After opening the data files which store the PBA Index values, you will create, or open, a file to store the pavement deterioration responsibilities:

Pavement Deterioration Proportions - Data File

1. Create a new proportions data file
2. Use an existing proportions data file
3. View a proportions data file
4. Return to Main Menu
Enter a choice:

A highlighted input-cell is displayed next to the **Enter a choice** label. To select an option, type the number to the left and press the **ENTER** key. These options are described below.

7.1 New Proportions Data File

The creation of a new data file has been explained in Section 5.1.1. Refer to that section, and follow the instructions in parenthesis for Section 7.1. If a file is created, refer to Section 7.5 for further instructions.

7.2 Existing Proportions Data File

The use of an existing data file has been explained in Section 5.1.2. Refer to that section, and follow the instructions in parenthesis for Section 7.2. If a file is open, refer to Section 7.2 for further instructions.

7.3 View a Proportions Data File

This option allows you to view the contents of a pavement deterioration proportions data file. To select this option type 3 and press the **ENTER** key.

> **Enter a choice:** 3
> **<press ENTER>**

Instructions to enter the file name will be displayed:

> **View Pavement Deterioration Proportions File**
>
> **Enter the name of the data file to be seen**
> **You can add a disk drive address (e.g. A:\), or a directory path (e.g. C:\PBA\ ). DO NOT ADD A FILE EXTENSION. (40 characters maximum).**

An input-cell is highlighted below this message. By default, the file will have the extension **.SHR**. If you typed in an extension, you will see an error message displayed at the bottom of the screen. After you enter the file name, one of two things may happen:

1. A file with that name is found. The data stored in the file will be displayed one record at a time. The information displayed will be:

Performance Based Approach
Oregon Department of Transportation
Highway Division

Record No.:
Highway Section:
District:
County:
Beginning Mile Post:
Ending Mile Post:
Region:

Pavement Deterioration Proportions:
Share of Load Related Factors:
Share of Non Load Related Factors:
Theoretical PBA Index:
Zero Maintenance PBA Index:

ENTER : continue **ESC : quit**

If you press the **ENTER** key, more records will be displayed. Pressing the **ESC** key will return you to the file menu of Section 7.

2. A file with that name is not found. An error message will be displayed at the bottom of the screen. Press the **ESC** key to return to the file menu of Section 7.

7.4 Return to Main Menu

This option will return you to the PBA Main Menu described in Section 2.

7.5 Pavement Deterioration Responsibilities

After you have successfully opened the data files to retrieve the PBA Index values, and created or opened a data file to store deterioration proportions, you will enter the identification number of the highway section for which the deterioration proportions will be computed. The data-input screen will be displayed:

Performance Based Approach
Oregon Department of Transportation
Highway Division

Enter the Highway ID:

An input-cell is displayed next to the ID word. Type in the identification number of the highway section, and then press the **ENTER** key. A message will be displayed in the middle of the screen:

Retrieving Highway Section Data
WAIT

The data for the highway section are being retrieved from the data files loaded on Section 7. These data are the Theoretical and Zero PBA Index values computed in Sections 3 and 4 respectively. If the data for the highway section are not found on any of these files, an error message will be displayed:

Highway Section: (highway ID) couldn't be found in:
(File Name)
Press any key to continue

and you will be asked to enter a new highway identification number. If you would like to quit, and return to the PBA Main Menu of Section 2, press the **ESC** key.

If data for the highway section are found, the results for the different pavement deterioration responsibilities will be displayed, together with other information previously entered or computed:

Highway Section:
District:
County:
Beginning Mile Post:
Ending Mile Post:
Region:

Pavement Deterioration Proportions:
Share of Load Related Factors:
Share of Non Load Related Factors:
Theoretical PBA Index:
Zero Maintenance PBA Index:

ENTER : continue

If you press the **ENTER** key, the data will be saved, and you will return to the Deterioration Proportions File Menu of Section 7.

8. Error Messages

If an error occurs during execution of PBA, a brief message describing the error will be displayed on the bottom of the screen. The errors that are trapped by PBA are briefly described below.

- **File Not Found:** File couldn't be found on the indicated path, or does not exist. The file name entered does not correspond to a file on the indicated drive.

- **Existing File:** The file already exists. A file having the same name exists on the indicated drive. Enter a different name.

- **Disk Format Error:** The disk in the indicated or default drive has not been formatted. Insert a formatted disk to continue. Refer to your DOS guide for instructions on how to format a floppy disk.

- **Bad File Name:** The file name entered contains invalid characters. Refer to your DOS guide for a list of legal characters that can be used in a file name.

- **Disk Full:** There is not enough free space on the drive to carry a file option; provide more free space and retry your selection.

- **Disk Error:** The disk is write-protected, or a bad diskette.

- **Disk Drive not Ready:** The door of a floppy disk drive is open, or there is not a disk in the indicated drive.

- **Path not Found:** The path you specified can not be found.

- **Missing Construction Year:** A date for the construction, or last rehabilitation of the highway section, was not provided when entering the highway identification data. Therefore, traffic analysis can no be performed. Enter the construction/rehabilitation year to continue with the execution of PBA.

- **Unidentified Error:** An error, not foreseen during the development of PBA, has occurred. Report the Error Code, the Error Address, and the activity you were performing when the error ocurred to the developers of PBA. The execution of PBA will automatically terminate.

9. Example

In this section you will go through some of the steps required to enter the data used by PBA to compute the pavement deterioration proportions for a highway section.

If you are using a hard disk, change to the PBA subdirectory you created when you first set up PBA. Run PBA by typing the following at the DOS prompt:

```
cd pba    <press ENTER>
PBA       <press ENTER>
```

After a few seconds, the PBA introductory screen will be displayed. Press the **ENTER** key to call up the PBA Main Menu.

Select option 1 of the menu, Compute Theoretical-Performance PBA Index. Place the highlighted cell on option 1, and then press the **ENTER** key. The file menu for the Theoretical-Performance PBA Index is displayed. Select option 1 to create a new data file:

```
Enter a choice:  1
<press ENTER>
```

Enter the name of the data file to be created:

```
              New Data File

Enter the name of the data file to be created
You can add a disk drive address (eg. A:), or a
directory path (eg. C:\PBA\ ).  DO NOT ADD A FILE
EXTENSION.  (40 characters maximum).

C:\pba\theory        <press ENTER>
```

Enter the data for the highway section:

```
Highway         :     0001
District        :     6
County          :     Douglas
Begin M.P.      :     98.92
End M.P.        :     103.95
Section EA      :     M0404432
EA/SJ           :     M6104430-002
Year            :     1976
Region          :     3
```

Select a performance measure. For this example use AASHTO Design Equation 1.2.1, therefore select option 1.

```
Enter a choice:  1
<press ENTER>
```

The traffic data will be input manually. Enter a **N** for the **Predicted Traffic** option, and a **Y** for the **Manual Input** option:

```
Predicted Traffic (y/n) :  n                  <press ENTER>
Manual Input (y/n) : y                        <press ENTER>
```

For this example, the Independent Performance variable will be Traffic. Enter the name and units of this variable:

```
             Prediction Values
Independent Variable of Predicted Performance

Enter the variable name and units:
   Name:         Traffic                      <press ENTER>
   Units:        ESAL                         <press ENTER>
   Variable ID:  ESAL                         <press ENTER>
```

Enter the number of data sets to be used:

```
Input the number of measurements
(maximum of 10 measurements) :    5           <press ENTER>
```

Enter the traffic data on the data-cells highlighted on the data-frame:

```
1976          <press ENTER>
0             <press ENTER>
1979          <press ENTER>
3570000       <press ENTER>
1981          <press ENTER>
5950000       <press ENTER>
1985          <press ENTER>
10710000      <press ENTER>
1988          <press ENTER>
12580000      <press ENTER>
```

Select a performance equation to be used. For this example we selected AASHTO equation 1.2.1 before. Select option 1 again:

```
Enter a choice:  1
<press ENTER>
```

Enter the structural data. For the Resilient Modulus use the R-value correlation:

```
Roadbed Resilient Modulus (Mr):
   Select a choice:  3                    <press ENTER>

   3. R-value correlation:     19         <press ENTER>

   3. R-value correlation:     19         Mr = 11545
```

Use the default values for Reliability and Standard Deviation:

```
Reliability (R, percent) =      <press ENTER>

Reliability (R, percent) =  95       Zr = -1.645

Standard Deviation (So) =       <press ENTER>
```

Enter the value for the Structural Number. Use option 2, Value from Design Specifications:

Select a choice: 2
<press ENTER>

Enter a value for Structural Number ==> SN = 5
<press ENTER>

To compute the PBA Index for the pavement structure, select option 1 of the Performance Measure menu now displayed:

Enter a choice: 1
<press ENTER>

After a few seconds, the Theoretical PBA Index will be displayed on the screen:

Highway ID : 0001
PBA Index : 11368584 PSI-ESAL
Do you want to plot the PSI-ESAL curve (y/n) ?

To view a plot of PSI vs. ESAL, type a **Y** and press the **ENTER** key. Otherwise, type a **N** and press the **ENTER** key to return to the PBA Highway Identification data screen. Return to the File Menu by pressing the **ESC** key. Select option 4 to return to the PBA Main Menu and continue with this example.

The second part of this example involves the determination of the Field-Peformance PBA Index for 3 pavement sections which form the highway section previously defined. Place the highlighted cell on option 2 of the PBA Main Menu, **Compute Field-Performance PBA Index**, and press the **ENTER** key. The Field-Performance menu will be displayed. Select option 1 to compute the field PBA Index for a pavement section:

Enter a choice: 1
<press ENTER>

This brings the Performance Equation - Data File Menu. Select option 1 to create a new file to store the information:

Enter a choice: 1
<press ENTER>

Enter the file name:

New Performance Equation File

Enter the name of the data file to be created
You can add a disk address (eg. A:) or a
directory path (eg. C:\PBA\ ). DO NOT ADD A FILE
EXTENSION. (40 characters maximum).

c:\pba\field
<press ENTER>

Select option 1 of the **Performance Equation** menu to create a new performance equation:

Enter a choice: 1
<press ENTER>

The performance equation we will use is:

PSI = -9.2556 + 10.3244 (log C) - 2.048 [(log C)^2]

where:

C = roadmeter counts per kilometer

This equation can be broken down into 3 sub-equations:

Sub-equation 1:	**-9.2556**
Sub-equation 2:	**10.3244 (log C)**
Sub-equation 3:	**-2.048 [(log C)^2]**

The screen to input the identification data for the equation is displayed. Enter the following:

Performance Measure	**:**	**PSI**	**<press ENTER>**
Performance Units	**:**	**PSI**	**<press ENTER>**
Performance Design Value	**:**	**4.2**	**<press ENTER>**
Maximum Performance Scale	**:**	**5**	**<press ENTER>**
Minimum Performance Scale	**:**	**0**	**<press ENTER>**
Pavement Section	**:**	**Test 1**	**<press ENTER>**
Equation ID number	**:**	**1**	**<press ENTER>**
Number of Subequations (2 minimum)	**:**	**3**	**<press ENTER>**

The screen to create the different variables is displayed. Enter the different subequations into PBA, and the values to be substituted in the variables. For sub-equation 1:

Variable Description	**:**	**constant**	**<press ENTER>**
Coefficient	**:**	**-9.2556**	**<press ENTER>**

Press the **E** key to end the creation of sub-equation 1. Enter the data for sub-equation 2:

Variable Description	**:**	**Roadmeter counts**	**<press ENTER>**
Coefficient	**:**	**10.3244**	**<press ENTER>**
Log(variable)	**:**		**<press ENTER>**

then, a lower case **x** will be displayed showing that the option is active:

Log(variable)	**:**	**x**

The cursor moves to the **Variable*(constant)** cell. Continue creating this sub-equation:

```
Variable*(constant)        :                    <press ENTER>
Constant^(variable)        :                    <press ENTER>
(Variable)^(exp)           :                    <press ENTER>
(Variable)+constant        :                    <press ENTER>
Variable Relation          :                    <press ENTER>
```

Press the ENTER key to input roadmeter counts. The data-input frame is displayed. Enter 5 measurements:

```
Number of measurements (max: 10) : 5
<press ENTER>

1976       <press ENTER>
1000       <press ENTER>
1979       <press ENTER>
1300       <press ENTER>
1981       <press ENTER>
1600       <press ENTER>
1985       <press ENTER>
1925       <press ENTER>
1988       <press ENTER>
2256       <press ENTER>
```

Press the E key to end creation of sub-equation 2, and start creating sub-equation 3. Break sub-equation 3 into three variables:

```
Variable 1:   (log C)
Variable 2:   (Variable 1)^2
Variable 3:   -2.048
```

For variable 1:

```
Variable Description:   Roadmeter Counts      <press ENTER>

  Coefficient       :                         <press ENTER>
  Log(variable)     :                         <press ENTER>
```

then,

```
  Log(variable)     :   x
```

The cursor moves down to the **Variable*(constant)** option:

```
Variable*(constant) :                    <press ENTER>
Constant^(variable) :                    <press ENTER>
(Variable)^(exp)    :                    <press ENTER>
(Variable)+constant :                    <press ENTER>
Variable Relation   :                    <press ENTER>
```

Press the **ENTER** key to input the values of roadmeter counts. The data-input frame is displayed. Enter 5 measurements:

```
Number of measurements (max: 10) : 5
<press ENTER>

1976       <press ENTER>
1000       <press ENTER>
1979       <press ENTER>
1300       <press ENTER>
1981       <press ENTER>
1600       <press ENTER>
1985       <press ENTER>
1925       <press ENTER>
1988       <press ENTER>
2256       <press ENTER>
```

Press the **ENTER** key to end the creation of variable 1, and to create variable 2. This variable will be a dummy variable used to evaluate (log C) to the second power.

```
Variable Description  :   dummy           <press ENTER>

  Coefficient         :                   <press ENTER>
  Log(variable)       :                   <press ↓>
  10^(variable)       :                   <press ↓>
  Ln(variable)        :                   <press ↓>
  e^(variable)        :                   <press ↓>
  Variable*(constant) :                   <press ENTER>
  Constant^(variable) :                   <press ENTER>
  (Variable)^(exp)    :                   <press ENTER>
  (Variable)+constant :                   <press ENTER>
  Variable Relation   : 211.20            <press ENTER>
```

Press the **ENTER** key to end the creation of variable 2, and to create variable 3. Enter the following:

Variable Description	:	dummy	<press ENTER>
Coefficient	:	-2.048	<press ENTER>
Log(variable)	:		<press ↓>
10^(variable)	:		<press ↓>
Ln(variable)	:		<press ↓>
e^(variable)	:		<press ↓>
Variable*(constant)	:		<press ENTER>
Constant^(variable)	:		<press ENTER>
(Variable)^(exp)	:		<press ENTER>
(Variable)+constant	:		<press ENTER>
Variable Relation	:	90	<press ENTER>

Press the **E** key to finish creating sub-equation 3. The next step is to define the relationship between the sub-equations. The different options are displayed on the screen. For relation **A**, which defines the relation between sub-equations 1 and 2, press the **ENTER** key when the highlighted option-cell is on the + option. For relation **B**, which defines the relation between **A** and sub-equation 3, press the **ENTER** key when the highlighted option-cell is on the + option.

After a few seconds, the screen used to enter data for the independent performance variable will be displayed. Enter the data for traffic so that the Field-PBA Index can be computed.

Field Measurements
Independent Variable of Field Performance Equation

Enter the variable name and units:

Pavement Section ID:	**Test 1**	<press ENTER>
Variable Name:	**Traffic**	<press ENTER>
Variable Units:	**ESAL**	<press ENTER>
Variable ID:	**ESAL**	<press ENTER>

Enter the number of measurements. These measurements are the amounts of traffic at the time the roadmeter counts were made:

```
Input the number of measurements
(maximum of 10 measurements)    : 5            <press ENTER>
```

Enter the traffic data in the data-frame displayed on the bottom of the screen:

```
1976        <press ENTER>
0           <press ENTER>
1979        <press ENTER>
3570000     <press ENTER>
1981        <press ENTER>
5950000     <press ENTER>
1985        <press ENTER>
10710000    <press ENTER>
1988        <press ENTER>
12580000    <press ENTER>
```

After entering the traffic data, the Field-PBA Index will be displayed on the screen:

```
Pavement ID:  Test 1
PBA Index = 17271904  PSI * ESAL
Do you want to plot the PSI * ESAL curve (y/n) ?
```

To view a plot of PSI vs. ESAL, type **Y**, otherwise type **N** to return to the PBA Index : Field Performance menu. Enter the data for the other two pavement sections.

For the second pavement section, select option 1 to compute the Field PBA Index:

```
Enter a choice: 1
<press ENTER>
```

The performance equation data file menu is displayed. Select option 2 to use the same data file as for the previous pavement section. Instructions to enter the data file name are displayed:

```
        Existing Performance Equation File
Enter the name of the data file to be opened
You can enter a disk drive address (eg. A:), or a
directory path (C:\PBA\ ).  DO NOT ENTER A FILE
EXTENSION.  (40 characters maximum).

c:\pba\field
<press ENTER>
```

A warning message is displayed on the bottom of the screen:

```
C:\PBA\FIELD.FEQ already exists.
Do you want to overwrite/delete it (y/n) ?
Do you want to add new information (y/n) ?
```

Enter the following:

```
Do you want to overwrite/delete it (y/n) ?  n    <press ENTER>
Do you want to add new information (y/n) ?  y    <press ENTER>
```

The performance equation menu will be displayed. Select option 2 to use the same performance equation that was created for the first pavement section:

```
Enter a choice: 2
<press ENTER>
```

The screen to enter identification data for the equation is displayed. Enter the following:

```
Performance Measure       :  PSI         <press ENTER>
Performance Units         :  PSI         <press ENTER>
Performance Design Value  :  4.2         <press ENTER>
Maximum Performance Scale :  5           <press ENTER>
```

Minimum Performance Scale : 0 <press ENTER>
Pavement Section : Test 2 <press ENTER>
Equation ID number : 1 <press ENTER>
Number of Sub-equations
(2 minimum) : 3 <press ENTER>

After a few seconds, the data-input frame will be displayed. Enter the values for the roadmeter counts variables of sub-equations 2 and 3. (You need to enter the same data twice).

Number of measurements (max: 10) : 5 <press ENTER>

1976 <press ENTER>
1000 <press ENTER>
1980 <press ENTER>
1500 <press ENTER>
1982 <press ENTER>
2000 <press ENTER>
1985 <press ENTER>
2500 <press ENTER>
1988 <press ENTER>
2700 <press ENTER>

Next, define the relationship between the sub-equations. For relation **A**, press the **ENTER** key when the highlighted option-cell is on the + option. For relation B, press the **ENTER** key when the highlighted option-cell is on the + option. After a few seconds, you will be requested to enter the independent variable for evaluating the Field-PBA Index. The data-input screen is displayed:

Field Measurements
Independent Variable of Field Performance Equation

Enter the variable name and units:
Pavement Section ID : Test 2 <press ENTER>
Variable Name: Traffic <press ENTER>
Variable Units: ESAL <press ENTER>
Variable ID: ESAL <press ENTER>

Enter the number of measurements:

Input the number of measurements
(maximum of 10 measurements) : 5 **<press ENTER>**

An equal number of highlighted cells are displayed on the data-input frame. Enter the following measurements:

1976	**<press ENTER>**
0	**<press ENTER>**
1980	**<press ENTER>**
3600000	**<press ENTER>**
1982	**<press ENTER>**
6200000	**<press ENTER>**
1985	**<press ENTER>**
11200000	**<press ENTER>**
1988	**<press ENTER>**
12580000	**<press ENTER>**

After a few seconds, the Field PBA Index will be displayed:

Pavement ID: Test 2
PBA Index: 20060186 PSI * ESAL
Do you want to plot PSI - ESAL curve (y/n) ?

To view a plot of PSI vs. ESAL, enter **Y** and press the **ENTER** key. Enter **N** to return to the PBA Index : Field Performance menu to continue with this example. For the third pavement section, follow the same procedure as for the second pavement section. Use equation 1, and the following values:

Roadmeter Counts:

Year	**Measure**
1976	**1000**
1979	**1500**
1982	**2000**
1986	**2450**
1988	**2700**

And for the traffic data:

Year	ESAL
1976	0
1979	3010000
1982	9000000
1986	12580000
1988	14100000

The field PBA Index will be displayed:

Pavement Section: Test 3
PBA Index: 22124156 PSI * ESAL
Do you want to plot PSI * ESAL curve (y/n) ?

Once you have computed the Field-PBA Index for the pavement sections, return to the PBA Index : Field Performance menu. The next step in computing pavement deterioration responsibilities is to estimate the PBA Index for Zero Level of Routine Maintenance. Select option 2 of the menu:

Enter a choice: 2
<press ENTER>

The Performance Equation Data File menu will be displayed. Select option 2 to use the data file created previously. Enter the name of the file:

c:\pba\field
<press ENTER>

Create a file to store the Zero-PBA Index, and the data used to compute the index. Select option 1 of the Total Pavement Deterioration-Data File menu:

Enter a choice: 1
<press ENTER>

Enter the name of the file:

c:\pba\zero
<press ENTER>

Enter the identification number of the highway section for which the Total-PBA Index is being computed:

PBA Index
Zero Level of Routine Maintenance
Enter the Highway Section ID: 0001 <press ENTER>

The screen to input the identification data for the pavement sections forming the highway section is displayed. First, enter the number of sections created before:

Input the number of Pavement Sections
(minimum : 2 maximum : 20) : 3 <press ENTER>

Enter the ID of each pavement section:

test 1 <press ENTER>
test 2 <press ENTER>
test 3 <press ENTER>

After a few seconds, the screen to input the measure of level of routine maintenance will be displayed. For this example, the measure is the Average Annual Routine Maintenance Expenditure per Lane-Mile, and the units are Dollars. Enter the name and units for this measure:

PBA Index
Zero Level of Routine Maintenance
Enter the measure name and units:
Measure Name: Avg. Cost l/m <press ENTER>
Measure Units: Dollars <press ENTER>

A data-input frame with the ID of the pavement sections is displayed. Enter the measure for each section:

Test 1:
350 <press ENTER>

Test 2:
210 <press ENTER>

Test 3:
90 <press ENTER>

After a few seconds the results are displayed on the screen. The PBA Index for a Zero Level of Routine Maintenance is shown in the PBA Index Intercept cell:

PBA Index Intercept: 23869778.55

Other statistical information is also shown. Press the **ENTER** key to view a plot of the data, and the regression line. Press any key to return to the PBA Index : Field Performance menu. Select option 3 to return to the PBA Main Menu.

Finally, to compute the pavement deterioration responsibilities, select option 4 of the main menu. Place the highlighted cell on option 4 and then press the **ENTER** key. You will be requested to enter the name of the data files which store the data needed.

First, enter the name of the data file that stores the Theoretical-PBA Index:

```
c:\pba\theory
<press ENTER>
```

Then, enter the name of the file that stores the Zero-PBA Index:

```
c:\pba\zero
<press ENTER>
```

The Pavement Deterioration Data File menu will be displayed next. Create a file that will save the pavement deterioration responsibilities, select option 1:

```
Enter a choice: 1
<press ENTER>
```

Enter the name of the file:

```
New Pavement Deterioration Proportions File

Enter the name of the data file to be created
You can add a disk drive address (eg. A:), or a
directory path (eg. C:\PBA\ ).  DO NOT ADD A FILE
EXTENSION.  (40 characters maximum).

c:\pba\share
<press ENTER>
```

Enter the identification of the highway section:

```
    Performance Based Approach
Oregon Department of Transportation
        Highway Division

Enter the highway ID:  0001                    <press ENTER>
```

The following results will be displayed:

Performance Based Approach
Oregon Department of Transportation
Highway Division

Highway Section:	0001
District :	6
County :	Douglas
Beginning Mile Post:	98.92
Ending Mile Post:	103.95
Region :	3

Pavement Deterioration Proportions:

Share of Load Related Factors :	0.7257
Share of Non Load Related Factors :	0.2743

Theoretical PBA Index:	11368584	PSI - ESAL
Zero Maintenance PBA Index:	22124156	PSI - ESAL

Therefore, the pavement deterioration responsibility of Load Related Factors is 72.57 %, and the pavement deterioration responsibility due to Non-Load Related Factors is 27.43 %.

Appendix A
PBA Source Code

AVAILABLE IN HARD COPY (220 PAGES)
OR ON DISK UPON REQUEST

WRITE TO:

OREGON STATE HIGHWAY DIVISION
RESEARCH UNIT
800 AIRPORT RD., SE
SALEM, OR 97310

Appendix A.1
PBA.BAS

```
'-----------------------------------------------------------------------------
'
'                              PBA.BAS
'
'                    "Performance Based Approach"
'                     "Field Performance Module"
'The following program was coded using Turbo BASIC version 1.0.
'This module is the main menu of the Performance Based Approach, and it is also
'used for the determination of the total amount of damage due to load-related
'and non-load-related factors.  Refer to section 2.2.3 of Report No. 1,
'"Effects of Environmental Factors on Pavement Deterioration".
'-----------------------------------------------------------------------------
    $STACK 4096  'increase the stack size
    GOSUB Graph  :  SCREEN 0
    CALL Monitor(VideoMode%)  'find out if color or monochrome
    CALL Frame    'create the introductory screen
    COLOR ColArr%(15),ColArr%(1)
    LOCATE 5,11,0
    PRINT "Effect of Environmental Factors on Pavement Deterioration"
    LOCATE 7,38,0   :  PRINT "for"
    LOCATE 9,22,0   :  PRINT "Oregon Department of Transportation"
    LOCATE 11,39,0  :  PRINT "by"
    LOCATE 13,32,0  :  PRINT "Gustavo Ordonez"
    LOCATE 14,26,0  :  PRINT "Graduate Research Assistant"
    LOCATE 15,38,0  :  PRINT "and"
    LOCATE 16,31,0  :  PRINT "Dr. Ted S. Vinson"
    LOCATE 17,29,0  :  PRINT "Principal Investigator"
'note: Gustavo Ordonez may be located at:
'                      4 avenida 43-85 zona 12
'                      Colonia Monte Maria III
'                      Guatemala, Guatemala
'                      telephone: 763107
    LOCATE 19,24,0  :  PRINT "Department of Civil Engineering"
    LOCATE 20,28,0  :  PRINT "Oregon State University"
    LOCATE 21,28,0  :  PRINT "Corvallis, Oregon 97331"
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 24,31,0  :  PRINT " ENTER : continue ";
    DO : LOOP UNTIL INKEY$ = CHR$(13)
'-----------------------------------------------------------------------------
'this routine creates the main menu, display the screen
      DO
        CALL MenuFrame
        'define the routine to trap errors
        ON ERROR GOTO ErrorTrap
        DIM DYNAMIC PBAmenu$(1:5),PBAmsg$(1:15)
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 23,22,0 :  PRINT " " + CHR$(25) + " : Down ";
        LOCATE 23,33,0 :  PRINT " " + CHR$(24) + " : Up ";
        LOCATE 23,42,0 :  PRINT " ENTER : Select ";
140     'initialize the arrays
```

```
PBAmenu$(1) = " 1. Compute Theoretical-Performance PBA Index "
PBAmenu$(2) = " 2. Compute Field-Performance PBA Index "
PBAmenu$(3) = " 3. PBA Index for Zero-Level of Routine Maintenance "
PBAmenu$(4) = " 4. Pavement Deterioration Proportions "
PBAmenu$(5) = " 5. Exit to DOS "
RESTORE 501
FOR RdMsg% = 1 TO 15
  READ PBAmsg$(RdMsg%)
NEXT RdMsg%
'initialize pointers
PBA% = 1 : PBArow% = 10 : Help$ = "N"
'display menu screen
COLOR ColArr%(15),ColArr%(1)
LOCATE 3,34,0 : PRINT "PBA Main Menu"
LOCATE 6,22,0 : PRINT "Oregon Department of Transportation"
LOCATE 7,32,0 : PRINT "Highway Division"
LOCATE 8,27,0 : PRINT "Performance Based Approach"
'display the PBA options
FOR I% = 1 TO 5
  LOCATE 9+I%,15,0
  PRINT PBAmenu$(I%)
NEXT I%
COLOR ColArr%(12),ColArr%(15)
LOCATE 10,15,0  :  PRINT USING "&"; PBAmenu$(1)
CALL PBAmenu(PBA%)
DO
  OnHelp$ = "MAIN"
  DO  :  LOOP UNTIL INSTAT
  Choice$ = INKEY$
  SELECT CASE Choice$
    CASE CHR$(13)                'trap ENTER
      LOCATE 10,15,0
      ERASE PBAmenu$,PBAmsg$
      SELECT CASE PBA%
        CASE 1              'theoretical performance
          'save the video card for use in PBAASHTO.EXE
          OPEN "PBAVIDEO.DAT" FOR OUTPUT AS # 3
          PRINT # 3, USING "##"; VideoGraph%
          CLOSE # 3
          FileErr$ = "PBAASHTO"
          SHELL "PBAASHTO"
          Help$ = "S"
        CASE 2,3            'field performance, zero level of maintenance
          CALL PBAfield
          Help$ = "S"
        CASE 4              'pavement deterioration proportions
          CALL PBAshare
          Help$ = "S"
        CASE 5              'exit to DOS, program termination
          COLOR ColArr%(15),ColArr%(1)
          CLS
          SYSTEM
```

```
                END SELECT
              CASE ELSE
                SELECT CASE MID$(Choice$,2,1)
                  CASE CHR$(59)
                    CALL PBAhelp(OnHelp$)
                  CASE CHR$(72),CHR$(80)              'trap arrow keys
220                 COLOR ColArr%(15),ColArr%(1)
                    LOCATE PBArow%,15,0
                    PRINT USING "&"; PBAmenu$(PBA%)
                    IF MID$(Choice$,2,1) = CHR$(80) THEN
                      INCR PBArow% : INCR PBA%
                    ELSE
                      DECR PBArow% : DECR PBA%
                    END IF
                    IF PBArow% > 14 THEN
                      PBArow% = 10 : PBA% = 1
                    ELSEIF PBArow% < 10 THEN
                      PBArow% = 14 : PBA% = 5
                    END IF
                    COLOR ColArr%(12),ColArr%(15)
                    LOCATE PBArow%,15,0
                    PRINT USING "&"; PBAmenu$(PBA%)
                    CALL PBAmenu(PBA%)
                END SELECT
            END SELECT
          LOOP UNTIL Help$ = "S"
300     LOOP   'going back to main menu
'.....................................................................
'display the GAP messages
SUB PBAmenu(PBA%)
320     SHARED ColArr%(),PBAmsg$()
        CALL CleanMsg(18,20)
        Msg% = (PBA% * 3) - 3
        COLOR ColArr%(14),ColArr%(1)
        FOR MsgRw% = 1 TO 3
          LOCATE (MsgRw%+17),10,0
          PRINT USING "&"; PBAmsg$(Msg%+MsgRw%)
        NEXT MsgRw%
END SUB
'.....................................................................
'this routine cleans the screen
SUB CleanMsg(ClStart%,ClEnd%)
10700 SHARED ColArr%()
        COLOR ColArr%(15),ColArr%(1)
        FOR ClScr% = ClStart% TO ClEnd%
          LOCATE ClScr%,7,0
          PRINT STRING$(66,0)
        NEXT ClScr%
        COLOR ColArr%(12),ColArr%(15)
END SUB
'---------------------------------------------------------------------
'create the frame on the screen
```

```
SUB MenuFrame
11400 SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      CLS
      LOCATE 2,30,0   :  PRINT CHR$(201) + STRING$(18,205) + CHR$(187)
      LOCATE 3,5,0    :  PRINT CHR$(218) + STRING$(24,196) + CHR$(182) +_
                               STRING$(18,0) + CHR$(199) + STRING$(24,196) +_
                               CHR$(191)
      LOCATE 4,5,0    :  PRINT CHR$(179) + STRING$(24,0) + CHR$(200) +_
                               STRING$(18,205) + CHR$(188) + STRING$(24,0) +_
                               CHR$(179)
      FOR Frm% = 5 TO 15
        LOCATE Frm%,5,0
        PRINT CHR$(179) + STRING$(68,0) + CHR$(179)
      NEXT Frm%
      LOCATE 16,5,0   :  PRINT CHR$(195) + STRING$(68,196) + CHR$(180)
      FOR Frm% = 17 TO 21
        LOCATE Frm%,5,0
        PRINT CHR$(179) + STRING$(68,0) + CHR$(179)
      NEXT Frm%
      LOCATE 22,5,0   :  PRINT CHR$(195) + STRING$(68,196) + CHR$(180)
      LOCATE 23,5,0   :  PRINT CHR$(179) + STRING$(68,0) + CHR$(179);
      LOCATE 24,5,0   :  PRINT CHR$(192) + STRING$(68,196) + CHR$(217);
END SUB
'------------------------------------------------------------------------------
'PBA index for field performance
SUB PBAfield
      SHARED ColArr%(),Help$,OnHelp$,Xval!(),Yval!(),VideoGraph%,NewName$
'create the menu for the selection of a procedure
800   DO
        COLOR ColArr%(15),ColArr%(1)
        CLS
        CALL FieldScr
        LOCATE 8,27,0   : PRINT "Performance Based Approach"
        LOCATE 9,26,0   : PRINT "PBA Index: Field Performance"
        LOCATE 11,15,0 : PRINT "1. Compute Field Performance-PBA Index"
        LOCATE 12,18,0 : PRINT "for a Pavement Section"
        LOCATE 13,15,0
        PRINT "2. Compute PBA-Index for Zero Level of Routine"
        LOCATE 14,18,0 : PRINT "Maintenance for a Highway Section"
        LOCATE 15,15,0 : PRINT "3. Return to PBA Main Menu"
        LOCATE 17,20,0 : PRINT "Enter a choice :"
        COLOR ColArr%(1),ColArr%(10)
        DO
          Help$ = "N"   :  OnHelp$ = "FIELD"
          CALL MenuChoice(Choice$,17,38)
          SELECT CASE Choice$
            CASE CHR$(49)          'get equation and compute index
              DIM DYNAMIC Xval!(1:10),Yval!(1:10)
              CALL UserEqn("F")
              ERASE Xval!,Yval!
              Help$ = "X"
```

```
          CASE CHR$(50)
            'open the equation file to get PBA indexes
            CALL UserEqn("Y")
            OnHelp$ = "FIELD"
            IF Help$ <> "C" THEN
              'open the data file to save the results of regression analysis
              CALL RegFile
              IF Help$ = "S" THEN
                'get the pavement sections for the highway section
                CALL ZeroLevel
                IF Help$ <> "Q" THEN
                  'get the measure of level of routine maintenance
                  CALL Maintenance
                END IF
              END IF
            END IF
            Help$ = "X"
          CASE CHR$(51)
            Help$ = "M"
            EXIT LOOP
        END SELECT
      LOOP UNTIL Help$ = "X"
    LOOP UNTIL Help$ = "M"
END SUB
'------------------------------------------------------------------------
'open the data file for the regression results
SUB RegFile
      SHARED Help$,Viewing$,ColArr%(),OnHelp$,FileLra$,NewName$
      'create the menu for file name
      DO
        CALL FileScreen
        NewName$ = "N" : Help$ = "N"  : Viewing$ = "N" : OnHelp$ = "ZERO"
        LOCATE 4,20,0  :  PRINT "Total Pavement Deterioration - Data File"
        LOCATE 6,15,0  :  PRINT "1. Create a new Deterioration data File"
        LOCATE 7,15,0  :  PRINT "2. Use an existent Deterioration data File"
        LOCATE 8,15,0  :  PRINT "3. View a Deterioration data File"
        LOCATE 9,15,0  :  PRINT "4. Return to Previous Menu"
        LOCATE 10,15,0  :  PRINT " Enter a choice : ";
        DO
          Valid$ = "N"
          CALL MenuChoice(Choice$,10,34)
          SELECT CASE Choice$
            CASE CHR$(49),CHR$(50),CHR$(51),CHR$(52)
              Valid$ = "Y"
          END SELECT
        LOOP UNTIL Valid$ = "Y"
        FileChoice% = VAL(Choice$)
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 4,20,0 : PRINT SPACE$(45)
        COLOR ColArr%(12),ColArr%(15)
        SELECT CASE FileChoice%
          CASE 1   'to create a brand new pavement deterioration file
```

```
          LOCATE 4,23,0  :  PRINT " New Pavement Deterioration File "
          FileSpec$ = " created"
        CASE 2   ' to use an existent performance equation file
          LOCATE 4,21,0  :  PRINT " Existent Pavement Deterioration File "
          FileSpec$ = " opened"
        CASE 3
          LOCATE 4,23,0  :  PRINT " View Pavement Deterioration File "
          FileSpec$ = " seen"  :  Viewing$ = "Y"
        CASE 4
          Help$ = "C"
      END SELECT
      IF Help$ <> "C" THEN
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 6,13,0
        PRINT "Enter name of pavement deterioration file to be" + FileSpec$
        'check the existence of the file
        FileExt$ = ".LRA"
        CALL FiNaCheck(FileExt$,FileChoice%)
        IF NewName$ = "Y" THEN
          CALL CleanScr
        ELSE
          CLOSE # 3
          IF FileChoice% = 3 THEN
            CALL ViewLRA(FileLra$)
            CLOSE # 3
          ELSE
            Help$ = "S"
          END IF
        END IF
      END IF
    LOOP UNTIL INSTR(1,"CS",Help$) <> 0
END SUB
'---------------------------------------------------------------------------
'this routine computes the PBA Index for a Zero Level of Routine Maintenance
SUB ZeroLevel
    SHARED Help$,ColArr%(),Pavement$(),IxPBA!(),PaveID$(),PBAindex$,OnHelp$
    SHARED Highway$,NoData%,FileEqn$,PefUnits$,UnitPerf$
    OnHelp$ = "ZERO"
    'input the ID of the pavement sections
    CALL InputData("Z")
    IF Help$ <> "Q" THEN
      'retrieve the PBA index for the pavement sections
      CALL PvtScreen
      DIM DYNAMIC Pavement$(1:20),IxPBA!(1:20,1:2)
      'get the IDs for the pavement sections
      InPv% = 2 : NxPv% = 1 : RwPv% = 1
      FOR Gpv% = 1 TO 20
        SWAP InPv%,NxPv%
        Pavement$(Gpv%) = PaveID$(RwPv%,InPv%)
        IF InPv% = 2 THEN INCR RwPv%
      NEXT Gpv%
      ERASE PaveID$
```

```
CALL OpenFile(FileEqn$)
EndFile& = LOF(4)/412
Pointer% = 1
'check existence of the pavement sections
DO
  PvtSct$ = UCASE$(Pavement$(Pointer%))
  Lth% = LEN(PvtSct$)
  FOR PvtRec& = 1 TO EndFile&
    Exists$ = "Y"
    GET # 4, PvtRec&
    IF PvtSct$ = LEFT$(UCASE$(Highway$),Lth%) AND_
                  PBAindex$ <> STRING$(4,48) THEN
      UnitPerf$ = PefUnits$
      EXIT FOR
    END IF
    Exists$ = "N"
  NEXT PvtRec&
  IF Exists$ = "Y" THEN
    IxPBA!(Pointer%,1) = CVS(PBAindex$)
    'check the following
    INCR Pointer%
  ELSE
    'a pavement section does not exist
    LOCATE 24,20,0 : PRINT " ENTER : edit ";
    LOCATE 24,35,0 : PRINT " D : delete ";
    LOCATE 24,48,0 : PRINT " ESC : quit ";
    COLOR ColArr%(15),ColArr%(1)
    LOCATE 22,12,0 : PRINT SPACE$(40)
    LOCATE 23,30,0 : PRINT SPACE$(20)
    MsgC% = (50 - Lth%)\2
    LOCATE 22,MsgC%,0
    PRINT USING "Section: &  "; UCASE$(Pavement$(Pointer%));
    PRINT "could not be found ";
    DO
      DO : LOOP UNTIL INSTAT
      Choice$ = INKEY$
      Valid$ = "N"
      SELECT CASE Choice$
        CASE CHR$(13),CHR$(27),CHR$(68),CHR$(100)
          Valid$ = "Y"
      END SELECT
    LOOP UNTIL Valid$ = "Y"
    SELECT CASE Choice$
      CASE CHR$(13)
        LOCATE 23,25,0
        PRINT " Enter New ID: ";
        COLOR ColArr%(12),ColArr%(15)
        PRINT SPACE$(15);
        CALL InputString(StringIN$,23,42,12)
        IF Help$ <> "Q" THEN
          Pavement$(Pointer%) = StringIN$
          CALL PvtScreen
```

```
            END IF
          CASE CHR$(27)
            Help$ = "Q"
            ERASE Pavement$,IxPBA!
          CASE CHR$(68),CHR$(100)
            FOR MvPv% = Pointer% TO 19
              Pavement$(MvPv%) = Pavement$(MvPv%+1)
            NEXT MvPv%
            DECR NoData%
            CALL PvtScreen
        END SELECT
      END IF
      IF Help$ = "Q" THEN EXIT LOOP
    LOOP UNTIL Pointer% > NoData%
    CLOSE # 4
    IF Help$ <> "Q" THEN
      'save the sections for use in the Maintenance routine
      DIM DYNAMIC PaveID$(1:10,1:2)
      InPv% = 2 : NxPv% = 1 : RwPv% = 1
      FOR GpV% = 1 TO 20
        SWAP InPv%,NxPv%
        PaveID$(RwPv%,InPv%) = Pavement$(GpV%)
        IF InPv% = 2 THEN INCR RwPv%
      NEXT GpV%
    END IF
  END IF
END SUB
'.......................................................................
'print the wait screen
SUB PvtScreen
      SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 21,10,0 : PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      LOCATE 22,10,0 : PRINT CHR$(179) + SPACE$(58) + CHR$(179);
      LOCATE 23,10,0 : PRINT CHR$(179) + SPACE$(58) + CHR$(179);
      LOCATE 24,10,0 : PRINT CHR$(179) + SPACE$(58) + CHR$(179);
      LOCATE 25,10,0 : PRINT CHR$(192) + STRING$(58,196) + CHR$(217);
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 22,25,0 : PRINT " Retrieving Pavement Sections ";
      LOCATE 23,37,0 : PRINT " WAIT ";
END SUB
'-----------------------------------------------------------------------
'get the measure of level of routine maintenance for each pavement section
SUB Maintenance
      SHARED Pavement$(),IxPBA!(),ColArr%(),Help$,NoData%,PaveID$()
      SHARED Dat!(),HighwayNo$,OnHelp$,Incpt#,Slope#,CorCoef!,FileLra$
      IF NoData% > 1 THEN
        'get the measure of level of routine maintenance
        OnHelp$ = "ZERO"
        CALL InputData("M")
        IF Help$ = "Q" THEN
          ERASE PaveID$
```

```
        ELSE
          InPv% = 2 : NxPv% = 1 : RwPv% = 1
          FOR GpV% = 1 TO 20
            SWAP InPv%,NxPv%
            IxPBA!(GpV%,2) = Dat!(RwPv%,InPv%)
            IF InPv% = 2 THEN INCR RwPv%
          NEXT GpV%
          ERASE Dat!,PaveID$
          'do regression analysis
          CALL RegAnal(NoData%,IxPBA!())
          'save the results
          CALL PefWait("W")
          CALL SavePBA
          'plot the regression line
          CALL PlotReg
        END IF
      ELSE
        '2 or more data are required for regression
        CALL PvtScreen
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 22,22,0 : PRINT " Two or more pavement sections are ";
        LOCATE 23,22,0 : PRINT " required for Regression Analysis. ";
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 24,27,0 : PRINT " Press ESC to continue ";
        DO
          Quit$ = INKEY$
        LOOP UNTIL Quit$ = CHR$(27)
        ERASE PaveID$
      END IF
      ERASE IxPBA!,Pavement$
END SUB
'-------------------------------------------------------------------------
'regression analysis of data obtained in Maintenance routine
SUB RegAnal(NoData%,IxPBA!(2))
      SHARED ColArr%(),Pavement$(),X!(),Y!(),Ycalc!(),HighwayNo$,Performance$
      SHARED UnitPerf$,XvUnit$,MeasUnits$,Incpt#,Slope#,CorCoef!,unitPBA$
      DIM DYNAMIC X!(1:20),Y!(1:20),Ycalc!(1:20)
      'save the data in X and Y arrays
      FOR I% = 1 TO 20
        X!(I%) = IxPBA!(I%,2)
        Y!(I%) = IxPBA!(I%,1)
      NEXT I%
      CALL FitLine(X!(),Y!(),Ycalc!(),Incpt#,Slope#,NoData%,SigA#,SigB#,CorCoef!)
      'display the results of regression analysis
      CALL EqnScr(HighwayNo$)
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 4,20,0
      PRINT USING " Coefficient of Correlation :  ###.#### "; CorCoef!
      LOCATE 5,20,0
      PRINT USING " PBA Index Intercept :  #########.####### "; Incpt#
      LOCATE 6,20,0
      PRINT USING " Regression Line Slope : #########.####### "; Slope#
```

```
        ResFmt$ = "##  \           \  #########.######  #########.###### "
        ResFmt$ = ResFmt$ + "#########.######"
        Head$ = "No.   Pavement         PBA Index     Level of Routine    Calculated"
        SubH$ = "     Section ID                        Maintenance        PBA Index"
        LOCATE 9,5,0 : PRINT Head$
        LOCATE 10,5,0 : PRINT SubH$
        'reformat the units for displaying
        Units$ = CHR$(40)
        CALL Units(Units$,UnitPerf$)
        Units$ = Units$ + CHR$(45)
        CALL Units(Units$,XvUnit$)
        Units$ = Units$ + CHR$(41)
        UnCol% = 32 - (LEN(Units$) / 2)
        LOCATE 11,UnCol%,0 : PRINT USING "&"; Units$
        unitPBA$ = Units$
        Units$ = CHR$(40)
        CALL Units(Units$,MeasUnits$)
        Units$ = Units$ + CHR$(41)
        UnCol% = 49 - (LEN(Units$) / 2)
        LOCATE 11,UnCol%,0 : PRINT USING "&"; Units$
        'display the contents
        Pnt% = 1
        DO
          NxRw% = 13
          CALL CleanReg
          DO
            LOCATE  NxRw%,5,0
            PRINT USING ResFmt$; Pnt%,Pavement$(Pnt%),IxPBA!(Pnt%,1),_
                                 IxPBA!(Pnt%,2),Ycalc!(Pnt%)
            INCR Pnt%   :  INCR NxRw%
            IF NxRw% > 22 THEN EXIT LOOP
          LOOP UNTIL Pnt% > NoData%
          DO
            DO : LOOP UNTIL INSTAT
            Choice$ = INKEY$
            Valid$ = "N"
            SELECT CASE Choice$
              CASE CHR$(13)
                Valid$ = "Y"
              CASE CHR$(27)
                Valid$ = "Y"
                Pnt% = NoData% + 1
            END SELECT
          LOOP UNTIL Valid$ = "Y"
        LOOP UNTIL Pnt% > NoData%
        ERASE X!,Y!
END SUB
'..............................................................................
'get the units
SUB Units(Units$,VarUnit$)
        NoPo% = 0
        DO
```

```
        INCR NoPo%
        Units$ = Units$ + MID$(VarUnit$,NoPo%,1)
        IF LEN(Units$) > 17 THEN EXIT LOOP
      LOOP UNTIL MID$(VarUnit$,NoPo%+1,1) < CHR$(33) AND NoPo% > 2
END SUB
'.........................................................................
'fit a regression line through the data
SUB FitLine(X!(1),Y!(1),Ycalc!(1),Incpt#,Slope#,NoData%,SigA#,SigB#,CorCoef!)
      SHARED ColArr%()
      'initialize the additive variables
      SumX# = 0  : SumY# = 0
      SumX2# = 0 : SumY2# = 0
      SumXY# = 0
      FOR n% = 1 TO NoData%
        ValX# = X!(n%)
        ValY# = Y!(n%)
        SumX# = SumX# + ValX#
        SumY# = SumY# + ValY#
        SumXY# = SumXY# + (ValX# * ValY#)
        SumX2# = SumX2# + ((ValX#)^2)
        SumY2# = SumY2# + ((ValY#)^2)
      NEXT n%
      SXX# = SumX2# - ((SumX# * SumX#) / NoData%)
      SXY# = SumXY# - ((SumX# * SumY#) / NoData%)
      SYY# = SumY2# - ((SumY# * SumY#) / NoData%)
      Slope# = SXY# / SXX#
      Incpt# = (((SumX2# * SumY#) - (SumX# * SumXY#)) / NoData%) / SXX#
      CorCoef! = SXY# / (SQR(SXX# * SYY#))
      SEE# = SQR((SumY2# - (Incpt# * SumY#) - (Slope# * SumXY#))/(NoData%-2))
      SigB# = SEE# / SQR(SXX#)
      SigA# = SigB# * (SQR(SumX2# / NoData%))
      FOR Ev% = 1 TO NoData%
        Ycalc!(Ev%) = Incpt# + (Slope# * X!(Ev%))
      NEXT Ev%
END SUB
'.........................................................................
'clean the regression screen
SUB CleanReg
      SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      FOR I% = 12 TO 22
        LOCATE I%,5,0
        PRINT SPACE$(70)
      NEXT I%
END SUB
'-------------------------------------------------------------------------
'plot the regression line
SUB PlotReg
      SHARED Xval!(),Yval!(),Ycalc!(),IxPBA!(),Incpt#,OnGraph$,VideoGraph%
      SHARED Help$,NoData%,CorCoef!,ColArr%(),MaxLevel!,PlGu%(),Xlag%
      SHARED HighwayID$,MeasUnits$
      DIM DYNAMIC Xval!(1:20),Yval!(1:20)
```

```
'obtain the maximum value for level of maintenance
MaxLevel! = 0
Xmin! = 0
FOR Max% = 1 TO 20
  IF MaxLevel! <= IxPBA!(Max%,2) THEN MaxLevel! = IxPBA!(Max%,2)
  Xval!(Max%) = IxPBA!(Max%,2)
NEXT Max%
'determine the X scale
CALL Xexp(MaxLevel!,ExpX%,Xval!(),Xpoint!,20,ExpShw%)
XlagI% = 0 : XlagF% = 10
Xdiv% = 50 : Xdiv1% = 1
DIM DYNAMIC PlGu%(1:10,1:5)
'read the graphics data base
RESTORE 9900
FOR Rrow% = 1 TO 10
  FOR Rcol% = 1 TO 5
    READ PlGu%(Rrow%,Rcol%)
  NEXT Rcol%
NEXT Rrow%
'define the X scale
CALL Xscale(XlagI%,XlagF%,Xpoint!,Xscale%,Xspace%,Xview%,Xmax!,ExpX%)
'determine the Y scale
MaxPBA! = 0
Ymin! = 0
FOR Max% = 1 TO 20
  IF MaxPBA! <= IxPBA!(Max%,1) THEN MaxPBA! = IxPBA!(Max%,1)
  IF Ymin! >= IxPBA!(Max%,1) THEN Ymin! = IxPBA!(Max%,1)
  Yval!(Max%) = IxPBA!(Max%,1)
NEXT Max%
IF MaxPBA! < Incpt# THEN MaxPBA! = CSNG(Incpt#)
MaxPBA$ = STR$(MaxPBA!)
IF MaxPBA! < 1 THEN
  NoPo% = 1 : Flow% = -1 : Bound% = -10
ELSE
  NoPo% = 1 : Flow% = 1  : Bound% = 10
END IF
'determine the first non-zero value
DO
  INCR NoPo%
  FstNo$ = MID$(MaxPBA$,NoPo%,1)
LOOP UNTIL INSTR(1,"123456789",FstNo$) <> 0
FstNo% = VAL(FstNo$)
'maximum Y value
MaxY% = FstNo% + 1
'determine the exponent
FOR ExpY% = 0 TO Bound% STEP Flow%
  IF Flow% = 1 THEN
    IF (MaxPBA!/10^(ExpY%)) < 1 THEN
      DECR ExpY%
      EXIT FOR
    END IF
  ELSE
```

```
    IF (MaxPBA!/10^(ExpY%)) > 1 THEN EXIT FOR
  END IF
NEXT ExpY%
Ymax! = MaxY% * (10^(ExpY%))
'create the graphics
Help$ = "Q"
IF Xmax! >= MaxLevel! AND Xmax! > Xmin! THEN
  IF MaxPBA! > Ymin! THEN
    IF Ymin! >= 0 THEN
      OnGraph$ = "Y"
      SELECT CASE VideoGraph%
        CASE 16
          SCREEN 9      'EGA
        CASE 14
          SCREEN 8      'CGA color
        CASE 6,7
          SCREEN 2      'Hercules, CGA b&w
        CASE ELSE
          EXIT IF
      END SELECT
      IF VideoGraph% >= 14 THEN COLOR ColArr%(15),ColArr%(1)
      LOCATE 3,18,0 : PRINT "PBA Index-Zero Level of Routine Maintenance"
      LOCATE 4,25,0
      PRINT USING "PBA Index: +##########.#######"; Incpt#
      LOCATE 5,25,0  : PRINT USING "Correlation: +###.####"; CorCoef!
      LOCATE 7,27,0  : PRINT "Linear Regression Analysis"
      LOCATE 8,15,0  : PRINT USING "##.#^^^^"; Ymax!
      LOCATE 18,14,0 : PRINT USING "##.#^^^^"; Ymin!
      'print a label for y axis
      FOR I% = 1 TO 10
        I$ = MID$("PBA Index",I%,1)
        LOCATE (I%+8),13,0
        PRINT I$
      NEXT I%
      LOCATE 22,35,0
      PRINT USING "Highway ID: & "; HighwayID$
      LOCATE 24,30,0 : PRINT "Press any key to continue";
      LOCATE 20,45,0 : PRINT UCASE$(MeasUnits$);
      PRINT USING " (E+##)"; ExpShw%
      'print the values for each x division
      Xaxis% = 0
      FOR Xdash% = 23 TO (23+(Xspace% * Xlag%)) STEP Xspace%
        LOCATE 19,Xdash%,0
        PRINT Xaxis%
        INCR Xaxis%,Xdiv1%
      NEXT Xdash%
      'define the graphics screen
      CALL Graphics(VideoGraph%,Xview%,Xscale%,ColArr%(),"Y")
      'arrange the array from highest to lowest
      Limit% = 20
      DO
        Swaps% = 0
```

```
          FOR I% = 1 TO (Limit%-1)
            IF Xval!(I%) < Xval!(I%+1) THEN
              SWAP Xval!(I%),Xval!(I%+1)
              SWAP Yval!(I%),Yval!(I%+1)
              SWAP Ycalc!(I%),Ycalc!(I%+1)
              Swaps% = I%
            END IF
          NEXT I%
        LOOP WHILE Swaps%
        'create the graphics window
        WINDOW (Xmin!,Ymin!) - (Xmax!,Ymax!)
        'plot the values
        SELECT CASE VideoGraph%
          CASE 14,16
            FOR T% = 1 TO NoData%
              PSET (Xval!(T%),Yval!(T%)), ColArr%(15)
            NEXT T%
            'plot the correlation line
            PSET (Xval!(1),Ycalc!(1)), ColArr%(15)
            FOR T% = 2 TO NoData%
              PSET (Xval!(T%),Ycalc!(T%)), ColArr%(15)
              LINE (Xval!(T%),Ycalc!(T%))-(Xval!(T%-1),Ycalc!(T%-1)), ColArr%(
            NEXT T%
            PSET (Xmin!,Incpt#), ColArr%(15)
            LINE (Xmin!,Incpt#)-(Xval!(NoData%),Ycalc!(NoData%)), ColArr%(18)
          CASE ELSE
            FOR T% = 1 TO NoData%
              PSET (Xval!(T%),Yval!(T%))
            NEXT T%
            'plot the correlation line
            PSET (Xval!(1),Ycalc!(1))
            FOR T% = 2 TO NoData%
              PSET (Xval!(T%),Ycalc!(T%))
              LINE (Xval!(T%),Ycalc!(T%))-(Xval!(T%-1),Ycalc!(T%-1))
            NEXT T%
            PSET (Xmin!,Incpt#)
            LINE (Xmin!,Incpt#)-(Xval!(NoData%),Ycalc!(NoData%))
        END SELECT
        Help$ = "S"
      END IF
    END IF
  END IF
  'press any key to go back to the main menu routine after
  'plotting the curve, or after an error in graphics data
  IF Help$ = "Q" THEN
    CALL PefWait("N")
    LOCATE 13,27,0  :  PRINT "Press any key to continue"
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 12,16,0
    PRINT " Error in graphics input data, or graphics card "
    DO : LOOP UNTIL INSTAT
    Null$ = INKEY$
```

```
      ELSE
        DO : LOOP UNTIL INSTAT
        Null$ = INKEY$
        VIEW
        SCREEN 0
      END IF
      ERASE PlGu%,Xval!,Yval!,Ycalc!
      OnGraph$ = ""
END SUB
'............................................................................
'determine the values for the X scale
SUB Xexp(MaxLevel!,ExpX%,Xval!(1),Xpoint!,ArrSze%,ExpShw%)
      IF MaxLevel! < 1 THEN
        Flow% = -1 : Bound% = -10
      ELSE
        Flow% = 1  : Bound% = 10
      END IF
      FOR ExpX% = 0 TO Bound% STEP Flow%
        IF MaxLevel! > 1 THEN
          IF (MaxLevel! / (10^(ExpX%))) < 1 THEN
            DECR ExpX%
            ExpShw% = ExpX%
            EXIT FOR
          END IF
        ELSE
          IF (MaxLevel! / (10^(ExpX%))) > 1 THEN
            'transform the values of X for graphics purposes
            FOR Tr% = 1 TO ArrSze%
              Xval!(Tr%) = Xval!(Tr%) / (10^(ExpX%))
            NEXT Tr%
            MaxLevel! = MaxLevel! / (10^(ExpX%))
            ExpShw% = ExpX%
            ExpX% = 0
            EXIT FOR
          END IF
        END IF
      NEXT ExpX%
      Xpoint! = MaxLevel! / (10^(ExpX%))
END SUB
'---------------------------------------------------------------------------
'save the results of the regression analysis
SUB SavePBA
      SHARED Pavement$(),HighwayNo$,Performance$,Incpt#,Slope#,PBAunit$
      SHARED CorCoef!,PefUnits$,XvUnit$,MeasUnits$,IxPBA!(),MeasName$,unitPBA$
      SHARED HighwayID$,PerfName$,PfUnit$,VarXunit$,MntcUnit$,MntcName$
      SHARED TotalPBA$,Slope$,CorrCoef$,Pvt1$,Pvt2$,Pvt3$,Pvt4$,Pvt5$,Pvt6$
      SHARED Pvt7$,Pvt8$,Pvt9$,Pvt10$,Pvt11$,Pvt12$,Pvt13$,Pvt14$,Pvt15$
      SHARED Pvt16$,Pvt17$,Pvt18$,Pvt19$,Pvt20$,PBA1$,PBA2$,PBA3$,PBA4$
      SHARED PBA5$,PBA6$,PBA7$,PBA8$,PBA9$,PBA10$,PBA11$,PBA12$,PBA13$,PBA14$
      SHARED PBA15$,PBA16$,PBA17$,PBA18$,PBA19$,PBA20$,LvMt1$,LvMt2$,LvMt3$
      SHARED LvMt4$,LvMt5$,LvMt6$,LvMt7$,LvMt8$,LvMt9$,LvMt10$,LvMt11$
      SHARED LvMt12$,LvMt13$,LvMt14$,LvMt15$,LvMt16$,LvMt17$,LvMt18$
```

```
        SHARED LvMt19$,LvMt20$,FileLra$
        'open the data file
        CALL LRAopen(FileLra$)
        'save the results
        LSET HighwayID$ = HighwayNo$
        LSET PerfName$ = Performance$
        LSET PfUnit$ = PefUnits$
        LSET VarXunit$ = XvUnit$
        LSET MntcName$ = MeasName$
        LSET MntcUnit$ = MeasUnits$
        LSET TotalPBA$ = MKD$(Incpt#)
        LSET PBAunit$ = unitPBA$
        LSET Slope$ = MKD$(Slope#)
        LSET CorrCoef$ = MKS$(CorCoef!)
        LSET Pvt1$ = Pavement$(1)  : LSET Pvt2$ = Pavement$(2)
        LSET Pvt3$ = Pavement$(3)  : LSET Pvt4$ = Pavement$(4)
        LSET Pvt5$ = Pavement$(5)  : LSET Pvt6$ = Pavement$(6)
        LSET Pvt7$ = Pavement$(7)  : LSET Pvt8$ = Pavement$(8)
        LSET Pvt9$ = Pavement$(9)  : LSET Pvt10$ = Pavement$(10)
        LSET Pvt11$ = Pavement$(11)  : LSET Pvt12$ = Pavement$(12)
        LSET Pvt13$ = Pavement$(13)  : LSET Pvt14$ = Pavement$(14)
        LSET Pvt15$ = Pavement$(15)  : LSET Pvt16$ = Pavement$(16)
        LSET Pvt17$ = Pavement$(17)  : LSET Pvt18$ = Pavement$(18)
        LSET Pvt19$ = Pavement$(19)  : LSET Pvt20$ = Pavement$(20)
        LSET PBA1$ = MKS$(IxPBA!(1,1))  : LSET PBA2$ = MKS$(IxPBA!(2,1))
        LSET PBA3$ = MKS$(IxPBA!(3,1))  : LSET PBA4$ = MKS$(IxPBA!(4,1))
        LSET PBA5$ = MKS$(IxPBA!(5,1))  : LSET PBA6$ = MKS$(IxPBA!(6,1))
        LSET PBA7$ = MKS$(IxPBA!(7,1))  : LSET PBA8$ = MKS$(IxPBA!(8,1))
        LSET PBA9$ = MKS$(IxPBA!(9,1))  : LSET PBA10$ = MKS$(IxPBA!(10,1))
        LSET PBA11$ = MKS$(IxPBA!(11,1))  : LSET PBA12$ = MKS$(IxPBA!(12,1))
        LSET PBA13$ = MKS$(IxPBA!(13,1))  : LSET PBA14$ = MKS$(IxPBA!(14,1))
        LSET PBA15$ = MKS$(IxPBA!(15,1))  : LSET PBA16$ = MKS$(IxPBA!(16,1))
        LSET PBA17$ = MKS$(IxPBA!(17,1))  : LSET PBA18$ = MKS$(IxPBA!(18,1))
        LSET PBA19$ = MKS$(IxPBA!(19,1))  : LSET PBA20$ = MKS$(IxPBA!(20,1))
        LSET LvMt1$ = MKS$(IxPBA!(1,2))  : LSET LvMt2$ = MKS$(IxPBA!(2,2))
        LSET LvMt3$ = MKS$(IxPBA!(3,2))  : LSET LvMt4$ = MKS$(IxPBA!(4,2))
        LSET LvMt5$ = MKS$(IxPBA!(5,2))  : LSET LvMt6$ = MKS$(IxPBA!(6,2))
        LSET LvMt7$ = MKS$(IxPBA!(7,2))  : LSET LvMt8$ = MKS$(IxPBA!(8,2))
        LSET LvMt9$ = MKS$(IxPBA!(9,2))  : LSET LvMt10$ = MKS$(IxPBA!(10,2))
        LSET LvMt11$ = MKS$(IxPBA!(11,2))  : LSET LvMt12$ = MKS$(IxPBA!(12,2))
        LSET LvMt13$ = MKS$(IxPBA!(13,2))  : LSET LvMt14$ = MKS$(IxPBA!(14,2))
        LSET LvMt15$ = MKS$(IxPBA!(15,2))  : LSET LvMt16$ = MKS$(IxPBA!(16,2))
        LSET LvMt17$ = MKS$(IxPBA!(17,2))  : LSET LvMt18$ = MKS$(IxPBA!(18,2))
        LSET LvMt19$ = MKS$(IxPBA!(19,2))  : LSET LvMt20$ = MKS$(IxPBA!(20,2))
        SetNo& = (LOF(3) \ 605) + 1
        PUT # 3, SetNo&
        CLOSE # 3
END SUB
'.........................................................................
'open the total pavement deterioration file
SUB LRAopen(FileLra$)
        SHARED HighwayID$,PerfName$,PfUnit$,VarXunit$,MntcUnit$,MntcName$,PBAunit$
```

```
      SHARED TotalPBA$,Slope$,CorrCoef$,Pvt1$,Pvt2$,Pvt3$,Pvt4$,Pvt5$,Pvt6$
      SHARED Pvt7$,Pvt8$,Pvt9$,Pvt10$,Pvt11$,Pvt12$,Pvt13$,Pvt14$,Pvt15$
      SHARED Pvt16$,Pvt17$,Pvt18$,Pvt19$,Pvt20$,PBA1$,PBA2$,PBA3$,PBA4$
      SHARED PBA5$,PBA6$,PBA7$,PBA8$,PBA9$,PBA10$,PBA11$,PBA12$,PBA13$,PBA14$
      SHARED PBA15$,PBA16$,PBA17$,PBA18$,PBA19$,PBA20$,LvMt1$,LvMt2$,LvMt3$
      SHARED LvMt4$,LvMt5$,LvMt6$,LvMt7$,LvMt8$,LvMt9$,LvMt10$,LvMt11$
      SHARED LvMt12$,LvMt13$,LvMt14$,LvMt15$,LvMt16$,LvMt17$,LvMt18$
      SHARED LvMt19$,LvMt20$
      OPEN FileLra$ AS # 3 LEN = 605
      FIELD # 3, 25 AS HighwayID$, 25 AS PerfName$, 15 AS PfUnit$,_
                 15 AS VarXunit$, 15 AS MntcUnit$, 15 AS MntcName$,_
                 8 AS TotalPBA$, 8 AS Slope$, 4 AS CorrCoef$, 15 AS PBAunit$,_
                 15 AS Pvt1$, 15 AS Pvt2$, 15 AS Pvt3$, 15 AS Pvt4$,_
                 15 AS Pvt5$, 15 AS Pvt6$, 15 AS Pvt7$, 15 AS Pvt8$,_
                 15 AS Pvt9$, 15 AS Pvt10$, 15 AS Pvt11$, 15 AS Pvt12$,_
                 15 AS Pvt13$, 15 AS Pvt14$, 15 AS Pvt15$, 15 AS Pvt16$,_
                 15 AS Pvt17$, 15 AS Pvt18$, 15 AS Pvt19$, 15 AS Pvt20$,_
                 4 AS PBA1$, 4 AS PBA2$, 4 AS PBA3$, 4 AS PBA4$, 4 AS PBA5$,_
                 4 AS PBA6$, 4 AS PBA7$, 4 AS PBA8$, 4 AS PBA9$, 4 AS PBA10$,_
                 4 AS PBA11$, 4 AS PBA12$, 4 AS PBA13$, 4 AS PBA14$, 4 AS PBA15$,_
                 4 AS PBA16$, 4 AS PBA17$, 4 AS PBA18$, 4 AS PBA19$, 4 AS PBA20$,_
                 4 AS LvMt1$, 4 AS LvMt2$, 4 AS LvMt3$, 4 AS LvMt4$,_
                 4 AS LvMt5$, 4 AS LvMt6$, 4 AS LvMt7$, 4 AS LvMt8$,_
                 4 AS LvMt9$, 4 AS LvMt10$, 4 AS LvMt11$, 4 AS LvMt12$,_
                 4 AS LvMt13$, 4 AS LvMt14$, 4 AS LvMt15$, 4 AS LvMt16$,_
                 4 AS LvMt17$, 4 AS LvMt18$, 4 AS LvMt19$, 4 AS LvMt20$
END SUB
'--------------------------------------------------------------------------
'display the contents of a deterioration file
SUB ViewLra(FileLra$)
      SHARED HighwayID$,PerfName$,PfUnit$,VarXunit$,MntcUnit$,MntcName$,PBAunit$
      SHARED TotalPBA$,Slope$,CorrCoef$,Pvt1$,Pvt2$,Pvt3$,Pvt4$,Pvt5$,Pvt6$
      SHARED Pvt7$,Pvt8$,Pvt9$,Pvt10$,Pvt11$,Pvt12$,Pvt13$,Pvt14$,Pvt15$
      SHARED Pvt16$,Pvt17$,Pvt18$,Pvt19$,Pvt20$,PBA1$,PBA2$,PBA3$,PBA4$
      SHARED PBA5$,PBA6$,PBA7$,PBA8$,PBA9$,PBA10$,PBA11$,PBA12$,PBA13$,PBA14$
      SHARED PBA15$,PBA16$,PBA17$,PBA18$,PBA19$,PBA20$,LvMt1$,LvMt2$,LvMt3$
      SHARED LvMt4$,LvMt5$,LvMt6$,LvMt7$,LvMt8$,LvMt9$,LvMt10$,LvMt11$
      SHARED LvMt12$,LvMt13$,LvMt14$,LvMt15$,LvMt16$,LvMt17$,LvMt18$
      SHARED LvMt19$,LvMt20$,ColArr%()
      CALL LRAopen(FileLra$)
      CALL EqnScr(FileLra$)
      DIM DYNAMIC Cnt%(1:20)
      FOR I% = 1 TO 20
        Cnt%(I%) = I%
      NEXT I%
      ResFmt$ = "##  \          \  ##########.######  ##########.###### "
      Head$ =   "No.   Pavement         PBA Index     Level of Routine "
      SubH$ =   "      Section ID                         Maintenance    "
      RecNo& = LOF(3) \ 605
      FOR ViewReg& = 1 TO RecNo&
        GET # 3, ViewReg&
        COLOR ColArr%(15),ColArr%(1)
```

```
LOCATE 3,5,0  : PRINT USING "Record No.: ####"; ViewReg&
LOCATE 3,30,0 : PRINT USING "Highway ID: &"; HighwayID$
LOCATE 4,5,0  : PRINT USING "Performance: &"; LEFT$(PerfName$,25)
LOCATE 4,45,0 : PRINT USING "Perf. Units: &"; LEFT$(PfUnit$,15)
LOCATE 5,5,0  : PRINT USING "Maintenance Level: &"; LEFT$(MntcName$,15)
LOCATE 5,45,0 : PRINT USING "Units: &"; LEFT$(MntcUnit$,15)
LOCATE 6,5,0
PRINT USING "Zero PBA Index: +#########.#######"; CVD(TotalPBA$)
LOCATE 6,45,0 : PRINT USING "Units: &"; LEFT$(PBAunit$,15)
LOCATE 7,5,0
PRINT USING "Reg. Eqn. Slope: ########.######"; CVD(Slope$)
LOCATE 7,45,0
PRINT USING "Correlation Coefficient: +##.####"; CVS(CorrCoef$)
LOCATE 9,12,0 : PRINT Head$
LOCATE 10,12,0 : PRINT SubH$
LOCATE 12,12,0
PRINT USING ResFmt$; Cnt%(1),Pvt1$,CVS(PBA1$),CVS(LvMt1$)
LOCATE 13,12,0
PRINT USING ResFmt$; Cnt%(2),Pvt2$,CVS(PBA2$),CVS(LvMt2$)
LOCATE 14,12,0
PRINT USING ResFmt$; Cnt%(3),Pvt3$,CVS(PBA3$),CVS(LvMt3$)
LOCATE 15,12,0
PRINT USING ResFmt$; Cnt%(4),Pvt4$,CVS(PBA4$),CVS(LvMt4$)
LOCATE 16,12,0
PRINT USING ResFmt$; Cnt%(5),Pvt5$,CVS(PBA5$),CVS(LvMt5$)
LOCATE 17,12,0
PRINT USING ResFmt$; Cnt%(6),Pvt6$,CVS(PBA6$),CVS(LvMt6$)
LOCATE 18,12,0
PRINT USING ResFmt$; Cnt%(7),Pvt7$,CVS(PBA7$),CVS(LvMt7$)
LOCATE 19,12,0
PRINT USING ResFmt$; Cnt%(8),Pvt8$,CVS(PBA8$),CVS(LvMt8$)
LOCATE 20,12,0
PRINT USING ResFmt$; Cnt%(9),Pvt9$,CVS(PBA9$),CVS(LvMt9$)
LOCATE 21,12,0
PRINT USING ResFmt$; Cnt%(10),Pvt10$,CVS(PBA10$),CVS(LvMt10$)
DO
  DO : LOOP UNTIL INSTAT
  Choice$ = INKEY$
  SELECT CASE Choice$
    CASE CHR$(13)
      CALL CleanReg
      LOCATE 12,12,0
      PRINT USING ResFmt$; Cnt%(11),Pvt11$,CVS(PBA11$),CVS(LvMt11$)
      LOCATE 13,12,0
      PRINT USING ResFmt$; Cnt%(12),Pvt12$,CVS(PBA12$),CVS(LvMt12$)
      LOCATE 14,12,0
      PRINT USING ResFmt$; Cnt%(13),Pvt13$,CVS(PBA13$),CVS(LvMt13$)
      LOCATE 15,12,0
      PRINT USING ResFmt$; Cnt%(14),Pvt14$,CVS(PBA14$),CVS(LvMt14$)
      LOCATE 16,12,0
      PRINT USING ResFmt$; Cnt%(15),Pvt15$,CVS(PBA15$),CVS(LvMt15$)
      LOCATE 17,12,0
```

```
          PRINT USING ResFmt$; Cnt%(16),Pvt16$,CVS(PBA16$),CVS(LvMt16$)
          LOCATE 18,12,0
          PRINT USING ResFmt$; Cnt%(17),Pvt17$,CVS(PBA17$),CVS(LvMt17$)
          LOCATE 19,12,0
          PRINT USING ResFmt$; Cnt%(18),Pvt18$,CVS(PBA18$),CVS(LvMt18$)
          LOCATE 20,12,0
          PRINT USING ResFmt$; Cnt%(19),Pvt19$,CVS(PBA19$),CVS(LvMt19$)
          LOCATE 21,12,0
          PRINT USING ResFmt$; Cnt%(20),Pvt20$,CVS(PBA20$),CVS(LvMt20$)
        CASE CHR$(27)
          EXIT LOOP
      END SELECT
    LOOP UNTIL Choice$ = CHR$(13)
    IF Choice$ = CHR$(13) THEN
      DO : LOOP UNTIL INSTAT
      Choice$ = INKEY$
    END IF
    CALL ClScr(FileLra$)
    IF Choice$ = CHR$(27) THEN EXIT FOR
  NEXT ViewReg&
  ERASE Cnt%
END SUB
'----------------------------------------------------------------------------------
'the following subroutine plots in the screen the performance measure vrs
'the independent variable.
SUB PBAindex
      SHARED MaxESAL!,Xmin!,ESALu!(),XvalMax%,MaxW18%,Xval!(),Yval!()
      SHARED AreaBetween!,PBAindex$,HighwayNo$,PerformanceM$,UnitsX$,ColArr%()
      SHARED Xpoint!,Xdiv%,Xdiv1%,XlagI%,XlagF%,X0%,ExpX%,PSI!(),Ymax!,Ymin!
      SHARED PfDesign!,NoData%,Help$,ExpShw%
      MaxESAL! = 0        ' obtain the greatest traffic value
      Xmin! = 0
9495  FOR MaxW18% = 1 TO 10
        IF MaxESAL! <= ESALu!(MaxW18%) THEN
          MaxESAL! = ESALu!(MaxW18%)
          XvalMax% = MaxW18%
        END IF
        IF Xmin! >= ESALu!(MaxW18%) THEN Xmin! = ESALu!(MaxW18%)
        Xval!(MaxW18%) = ESALu!(MaxW18%)      'these are the values to be
        Yval!(MaxW18%) = PSI!(MaxW18%)        'plotted by PlotGraph
      NEXT MaxW18%
      IF AreaBetween! <> 0 THEN
        LSET PBAindex$ = MKS$(AreaBetween!)
        LOCATE 11,30,0
        PRINT USING "Pavement ID: & "; HighwayNo$; : PRINT " "
        LOCATE 12,11,0  :  PRINT SPACE$(55)
        LOCATE 12,10,0
        PRINT UCASE$(PerformanceM$) + CHR$(45) + MID$(UnitsX$,1,4) + " Index = ";
        COLOR ColArr%(12),ColArr%(15)
        'print the performance-? index
        IF AreaBetween! < 1 THEN
          PRINT USING " ##.######^^^^ "; AreaBetween!;
```

```
      ELSE
        PRINT USING " ########## "; AreaBetween!;
      END IF
      COLOR ColArr%(15),ColArr%(1)
      PRINT " " + UCASE$(PerformanceM$) + " *  " + MID$(UnitsX$,1,4)
      LOCATE 13,10,0
      PRINT "Do you want to plot the " + UCASE$(PerformanceM$);
      PRINT " vs " + MID$(UnitsX$,1,4) + " curve (y/n)? : "
      DO
        CALL MenuChoice(Choice$,13,64)
        COLOR ColArr%(15),ColArr%(1)
        SELECT CASE UCASE$(Choice$)
          CASE "Y"
            'determine the X scale
            CALL Xexp(MaxESAL!,ExpX%,Xval!(),Xpoint!,10,ExpShw%)
            Xdiv% = 50  : Xdiv1% = 1 : XlagI% = 0
            XlagF% = 10 : X0% = 0
            CALL PlotGraph
            EXIT LOOP
          CASE "N"
            EXIT LOOP
        END SELECT
      LOOP
      SCREEN 0
    END IF
    ERASE ESALu!,PSI!
END SUB
'-----------------------------------------------------------------------------
'the following subroutine plots the performance vrs independent variable curve
SUB PlotGraph
    SHARED Xmin!,ESALu!(),XvalMax%,MaxW18%,MaxESAL!,Xval!(),Yval!(),Xlag%
    SHARED AreaBetween!,PBAindex$,HighwayNo$,PerformanceM$,UnitsX$,ColArr%()
    SHARED Xpoint!,Xdiv%,Xdiv1%,XlagI%,XlagF%,X0%,ExpX%,Ymax!,Ymin!,ExpShw%
    SHARED VideoGraph%,PfDesign!,NoData%,Help$,OnGraph$,PlGu%(),VideoMode%
    'retrieve the graphics dbase
    'PlGu%( ,1)=Xview%; PlGu%( ,2)=Xscale%; PlGu%( ,3)=Xmax!;
    'PlGu%( ,4)=Dot!; PlGu%( ,5)=Xscale% for Hercules
    DIM DYNAMIC PlGu%(1:10,1:5)
    'read the database for graphics
    RESTORE 9900
    FOR Rrow% = 1 TO 10
      FOR Rcol% = 1 TO 5
        READ PlGu%(Rrow%,Rcol%)
      NEXT Rcol%
    NEXT Rrow%
    'define the X scale
    CALL Xscale(XlagI%,XlagF%,Xpoint!,Xscale%,Xspace%,Xview%,Xmax!,ExpX%)
    'determine if there is any error on graphics data
    Help$ = "Q"
    IF Xmax! >= Xval!(XvalMax%) AND Xmax! > Xmin! THEN
      IF Ymax! > Ymin! THEN
        IF Xmin! >= 0 AND Ymin! >= 0 THEN
```

```
OnGraph$ = "Y"  'flag for error routine
SELECT CASE VideoGraph%
  CASE 16
    SCREEN 9   'EGA
  CASE 14
    SCREEN 8   'CGA color
  CASE 6,7
    SCREEN 2   'Hercules, CGA monochrome
  CASE ELSE
    EXIT IF
END SELECT
IF VideoGraph% >= 14 THEN COLOR ColArr%(15),ColArr%(1)
LOCATE 3,16,0
PRINT UCASE$(PerformanceM$)+CHR$(45)+MID$(UnitsX$,1,4)+" Index = ";
'print the performance-? index
IF AreaBetween! < 1 THEN
  PRINT USING " ##.######^^^^ "; AreaBetween!;
ELSE
  PRINT USING " ##########.#### "; AreaBetween!;
END IF
PRINT " " + UCASE$(PerformanceM$) + " *  " + MID$(UnitsX$,1,4)
LOCATE 8,14,0  :  PRINT USING "###.#^^^^"; Ymax!
LOCATE 18,14,0 :  PRINT USING "##.#^^^^"; Ymin!
'print a label for y axis
FOR I% = 1 TO 10
  I$ = MID$(PerformanceM$,I%,1)
  LOCATE (I%+7),13,0
  PRINT I$
NEXT I%
LOCATE 22,35,0
PRINT USING "Pavement ID: & "; HighwayNo$
LOCATE 24,30,0 : PRINT "Press any key to continue";
LOCATE 20,53,0 : PRINT UCASE$(UnitsX$);
PRINT USING " (E+##)"; ExpShw%
'print on the screen the values for each X-axis division
ESALaxis% = X0%
FOR Xdash% = 23 TO (23+(Xspace%*Xlag%)) STEP Xspace%
  LOCATE 19,Xdash%,0
  PRINT ESALaxis%
  INCR ESALaxis%,Xdiv1%
NEXT Xdash%

'define the graphics screen
CALL Graphics(VideoGraph%,Xview%,Xscale%,ColArr%(),"N")
'the active screen will have world coordinates (Xmin!,Ymin!) as
'origin, and (Xmax!,Ymax!) in the upper-right corner
WINDOW (Xmin!,Ymin!) - (Xmax!,Ymax!)
'plot the values
SELECT CASE VideoGraph%
  CASE 14, 16
    'avoid graphics errors
    IF PfDesign! > 0 THEN
```

```
              LINE (Xmin!,PfDesign!)-(Xval!(XvalMax%),PfDesign!),ColArr%(10)
            END IF
            PSET (Xval!(1),Yval!(1)) ,ColArr%(15)
            FOR T% = 2 TO NoData%
              PSET (Xval!(T%),Yval!(T%)) ,ColArr%(15)
              LINE (Xval!(T%),Yval!(T%))-(Xval!(T%-1),Yval!(T%-1)),ColArr%(18)
            NEXT T%
          CASE ELSE
            IF PfDesign! > 0 THEN
              LINE (Xmin!,PfDesign!) - (Xval!(XvalMax%),PfDesign!)
            END IF
            PSET (Xval!(1),Yval!(1))
            FOR T% = 2 TO NoData%
              PSET (Xval!(T%),Yval!(T%))
              LINE (Xval!(T%),Yval!(T%)) - (Xval!(T%-1),Yval!(T%-1))
            NEXT T%
        END SELECT
        Help$ = "S"
      END IF
    END IF
  END IF
  'press any key to go back to the main data collection routine after
  'plotting the PSI-ESAL curve, or after an error in graphics data
  IF Help$ = "Q" THEN
    CALL PefWait$("N")
    LOCATE 13,27,0  :  PRINT "Press any key to continue"
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 12,16,0
    PRINT " Error in graphics input data, or graphics card "
    DO : LOOP UNTIL INSTAT
    Null$ = INKEY$
  ELSE
    DO : LOOP UNTIL INSTAT
    Null$ = INKEY$
    VIEW
  END IF
  ERASE PlGu%
  OnGraph$ = ""
END SUB
'.........................................................................
'define the X scale
SUB Xscale(XlagI%,XlagF%,Xpoint!,Xscale%,Xspace%,Xview%,Xmax!,ExpX%)
  SHARED PlGu%(),VideoMode%,MaxLevel!,Xlag%
  FOR Xlag% = XlagI% TO XlagF%
    'determine the number of divisions in the X-axis
    IF Xpoint! >= 0 AND Xpoint! <= Xlag% THEN
      'determine the number of pixels between each X-axis division
      IF VideoMode% = 7 THEN
        Xscale% = PlGu%(Xlag%,5)
        Xspace% = Xscale%/9
        Xview% = (PlGu%(Xlag%,5) * Xlag%) + 212
      ELSE
```

```
            Xscale% = PlGu%(Xlag%,2)
            Xspace% = Xscale%/8
            Xview% = PlGu%(Xlag%,1)
          END IF
          Xmax! =  PlGu%(Xlag%,3) * (10^(ExpX%-1))
          EXIT FOR
        END IF
9800  NEXT Xlag%
END SUB
'...........................................................................
'define the active graphics screen
SUB Graphics(VideoGraph%,Xview%,Xscale%,ColArr%(1),OnReg$)
      SELECT CASE VideoGraph%
        CASE 16,14,6      'EGA, CGA color, CGA b&w
          IF VideoGraph% = 16 THEN RESTORE 9950 ELSE RESTORE 9975
          DIM DYNAMIC ECga%(1:12)
          FOR Rga% = 1 TO 12
            READ ECga%(Rga%)
          NEXT Rga%
          FOR C% = 188 TO Xview% STEP Xscale%
            LINE (C%,ECga%(1)) - (C%,ECga%(2)) ,ColArr%(15)
          NEXT C%
          LINE (40,ECga%(3)) - (600,ECga%(4)) ,ColArr%(15),B
          LINE (44,ECga%(5)) - (596,ECga%(6)) ,ColArr%(15),B
          LINE (88,ECga%(7)) - (550,ECga%(8)) ,ColArr%(12),B
          IF OnReg$ = "Y" THEN
            'if plotting the regression line
            LINE (180,ECga%(9)) - (188,ECga%(9)) ,ColArr%(15)
            LINE (180,ECga%(1)) - (188,ECga%(1)) ,ColArr%(15)
          ELSE
            FOR L% = ECga%(9) TO ECga%(1) STEP ECga%(10)
              LINE (180,L%) - (188,L%) ,ColArr%(15)
            NEXT L%
          END IF
          LINE (188,ECga%(11)) - (188,ECga%(9)) ,ColArr%(15)
          LINE (Xview%,ECga%(1)) - (Xview%+8,ECga%(1))
          VIEW (188,ECga%(1)) - (Xview%,ECga%(9)),ColArr%(1)
          LINE (0,0) - (0,ECga%(12)) ,ColArr%(15)
          LINE - ((Xview%-180),ECga%(12)) ,ColArr%(15)
          ERASE ECga%
        CASE 7        'HERCULES
          FOR C% = 212 TO (Xview%) STEP Xscale%
            LINE (C%,245) - (C%,252)
          NEXT C%
          LINE (45,8) - (675,344) ,,B
          LINE (49,9) - (671,342) ,,B
          LINE (99,70) - (619,315) ,,B
          IF OnReg$ = "Y" THEN
            'if plotting the regression line
            LINE (203,105) - (212,105)
            LINE (203,245) - (212,245)
          ELSE
```

```
          FOR L% = 105 TO 245 STEP 28
            LINE (203,L%) - (212,L%)
          NEXT L%
        END IF
        LINE (212,98) - (212,105)
        LINE (Xview%,245) - (Xview%+9,245)
        VIEW (212,245) - (Xview%,105)
        LINE (0,0) - (0,140)
        LINE - ((Xview%-194),140)
    END SELECT
END SUB
'-----------------------------------------------------------------------
SUB PefWait(Mesg$)
      SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 10,8,0 : PRINT CHR$(201) + STRING$(62,205) + CHR$(187)
      FOR I% = 11 TO 14
        LOCATE I%,8,0 : PRINT CHR$(186) + SPACE$(62) + CHR$(186)
      NEXT I%
      LOCATE 15,8,0 : PRINT CHR$(200) + STRING$(62,205) + CHR$(188)
      IF Mesg$ = "Y" THEN
        LOCATE 12,23,0 : PRINT "The PBA Index is being computed"
        COLOR ColArr%(16),ColArr%(10)
        LOCATE 13,37,0 :  PRINT " WAIT "
        COLOR ColArr%(15),ColArr%(1)
      ELSEIF Mesg$ = "W" THEN
        LOCATE 12,22,0 : PRINT "The regression data are being saved"
        COLOR ColArr%(16),ColArr%(10)
        LOCATE 13,37,0 :  PRINT " WAIT "
        COLOR ColArr%(15),ColArr%(1)
      END IF
END SUB
'-----------------------------------------------------------------------
'display the introductory frame
SUB FieldScr
      LOCATE 7,10,0  : PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      LOCATE 8,10,0  : PRINT CHR$(179) + SPACE$(58) + CHR$(179)
      LOCATE 9,10,0  : PRINT CHR$(179) + SPACE$(58) + CHR$(179)
      LOCATE 10,10,0 : PRINT CHR$(195) + STRING$(58,196) + CHR$(180)
      FOR edge% = 11 TO 17
        LOCATE edge%,10,0
        PRINT CHR$(179) + SPACE$(58) + CHR$(179)
      NEXT edge%
      LOCATE 18,10,0 : PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
END SUB
'-----------------------------------------------------------------------
'link the BASIC programs which form PBAFIELD.BAS
      $INCLUDE "PBAEQN.BAS"
      $INCLUDE "PBASHARE.BAS"
      $INCLUDE "PBAFFILE.BAS"
      $INCLUDE "PBAFEDIT.BAS"
'-----------------------------------------------------------------------
```

```
'the messages for the PBA options are saved here
501    DATA "Compute Theoretical PBA Index for a highway section based,"
502    DATA "on a predictive performance equation (ie. AASHTO Equations"
503    DATA "1.2.1 or PP.26, or User's Equation)."
504    DATA "Compute Field PBA Index for a pavement section, based on"
505    DATA "field performance measurements, using user's pavement"
506    DATA "performance equation."
507    DATA "Compute Total PBA Index for a highway section based on field"
508    DATA "performance data of its pavement sections, and their respective"
509    DATA "measure of level of routine maintenance."
510    DATA "Compute pavement deterioration responsibilities of load and"
511    DATA "non-load related factors for a highway section."
512    DATA ""
513    DATA "Exit to DOS terminates the execution of PBA and returns"
514    DATA "control to DOS operating system."
515    DATA ""
'database for graphics routine
9900   DATA 516, 328,  10,  6, 369
       DATA 508, 160,  20, 10, 180
       DATA 500, 104,  30, 15, 117
       DATA 508,  80,  40, 25,  90
       DATA 508,  64,  50, 30,  72
       DATA 524,  56,  60, 40,  63
       DATA 524,  48,  70, 50,  54
       DATA 508,  40,  80, 60,  45
       DATA 476,  32,  90, 60,  36
       DATA 508,  32, 100, 70,  36
'database for EGA
9950   DATA 245, 252, 14, 344, 16, 342, 70, 315, 105, 28, 98, 140
'database for CGA
9975   DATA 140, 144,  8, 196,  9, 195, 40, 180,  60, 16, 56,  80
'----------------------------------------------------------------------------
'end of PBAFIELD.BAS
EndPBA:
      CLOSE
      END
'----------------------------------------------------------------------------
'this routine traps the errors that might ocurr during program execution
ErrorTrap:
      IF OnGraph$ = "Y" THEN SCREEN 0    'change to text screen
      SELECT CASE ERR
        CASE 5
          SELECT CASE ERL
            CASE 9780,9785  'checking the graphics card available
              SCREEN
              RESUME 9790
          END SELECT
      END SELECT
      DIM DYNAMIC ErMsg$(1:7)
      LOCATE 20,20,0   'turn cursor off
      'variables used in the ErrowWindow routine
      ErMsg$(4) = "Press ESC to continue" : ErMsg$(2) = "" : ErMsg$(5) = ""
```

```
 ErMsg$(3) = "ERROR : System"  :  InHelp$ = "N"
 SELECT CASE ERR
   CASE 33
     ErMsg$(1) = "File " + FileErr$ + " couldn't be found on"
     ErMsg$(2) = "the indicated path, or does not exist."
   CASE 58
     ErMsg$(1) = "The file " + FileErr$ + " already exists "
   CASE 53
     ErMsg$(1) = "File " + FileErr$ + " couldn't be found on"
     ErMsg$(2) = "the indicated path, or does not exist."
   CASE 57
     ErMsg$(1) = "The disk in the indicated or default drive has not"
     ErMsg$(2) = "been formatted, insert a formatted disk to continue"
   CASE 54,64
     ErMsg$(1) = "The file name entered contains invalid characters"
   CASE 61
     ErMsg$(1) = "There isn't enough free space on the drive to carry a"
     ErMsg$(2) = "file option; provide more free space and retry" +_
                 " your selection."
   CASE 70,72
     ErMsg$(1) = "The disk is write-protected, or bad diskette"
   CASE 71
     ErMsg$(1) = "The door of a floppy disk drive is open, or"
     ErMsg$(2) = "there is not disk in the indicated drive."
   CASE 76
     ErMsg$(1) = "The path you specified can not be found"
   CASE ELSE
     ErMsg$(1) = "Unidentified Error"
     ErMsg$(2) = "Error Code: " + STR$(ERR) +_
                "  Error Address: " + STR$(ERADR) +_
                "  Error Line: " + STR$(ERL)
     ErMsg$(4) = "Report code and address to PBA developers"
END SELECT
'display the error window
CLOSE
CALL MsgWindow(17,5,6,70,FNATTR%(ColArr%(14),ColArr%(4)),1,0,0)
ERASE ErMsg$
SELECT CASE ERL
  CASE 180                    'PBAASHTO.TBC was not found
    RESUME NEXT
  CASE 755                    'error on free disk space
    CLOSE # 1
    NoSpace$ = "Y"
    RESUME NEXT
  CASE 17300,17310,17320,17330,17340,17350     'error on data files
    NewName$ = "Y"
    Help$ = "F"
    RESUME NEXT
  CASE 17330                  'existent file
    RESUME NEXT
  CASE 4050                   'error on Help file
    Help$ = "H"
```

```
    RESUME NEXT
  CASE ELSE                    'something unexpected went wrong
    RESUME EndPBA
END SELECT
'--------------------------------------------------------------------------------
```

Appendix A.2
PBASHARE.BAS

```
'---------------------------------------------------------------------------
'
'                              PBASHARE.BAS
'
'                 Pavement Deterioration Proportions Module
'The following program was coded using Turbo BASIC version 1.1.  This module
'computes the share of load and non-load related factors on pavement
'deterioration.
'---------------------------------------------------------------------------
'turn key trapping off
      $EVENT OFF
'---------------------------------------------------------------------------
SUB PBAshare
      SHARED Help$,Viewing$,ColArr%(),OnHelp$,FileLra$,FileDat$,FileErr$
      SHARED FileShare$,Shares$,NewName$
      CALL FileScreen
      DO
        CALL CleanScr
        OnHelp$ = "SHARE"
        NewName$ = "N" : Help$ = "N" : Viewing$ = "Y" : Shares$ = "Y"
        'retrieve the theoretical PBA index
        LOCATE 4,28,0 : PRINT "Theoretical PBA Index"
        LOCATE 6,13,0
        PRINT "Enter name of data file to retrieve data"
        FileExt$ = ".DAT"
        CALL FiNaCheck(FileExt$,2)
        IF NewName$ <> "Y" THEN
          CLOSE # 1
          'retrieve field PBA Index
          LOCATE 4,19,0 : PRINT "PBA Index-Zero Level Routine Maintenance"
          LOCATE 6,13,0
          PRINT "Enter name of deterioration file to retrieve data"
          FileExt$ = ".LRA"
          CALL FiNaCheck(FileExt$,2)
          IF NewName$ <> "Y" THEN
            DO
              CLOSE # 3
              'create or open file to store the different proportions
              CALL FileScreen
              NewName$ = "N" : Help$ = "N" : Viewing$ = "N"
              LOCATE 4,17,0
              PRINT " Pavement Deterioration Proportions - Data File "
              LOCATE 6,15,0 : PRINT "1. Create a new Proportions data File"
              LOCATE 7,15,0 : PRINT "2. Use an existent Proportions data File"
              LOCATE 8,15,0 : PRINT "3. View a Proportions File"
              LOCATE 9,15,0 : PRINT "4. Return to Main Menu"
              LOCATE 10,15,0 : PRINT " Enter a choice : ";
              DO
                Valid$ = "N"
                CALL MenuChoice(Choice$,10,34)
                SELECT CASE Choice$
                  CASE CHR$(49) TO CHR$(52)
```

```
                Valid$ = "Y"
            END SELECT
          LOOP UNTIL Valid$ = "Y"
          FileChoice% = VAL(Choice$)
          COLOR ColArr%(12),ColArr%(15)
          SELECT CASE FileChoice%
            CASE 1
              'to create a brand new pavement deterioration file
              LOCATE 4,17,0
              PRINT "  New Pavement Deterioration Proportions File  "
              FileSpec$ = " created"
            CASE 2
              ' to use an existent performance equation file
              LOCATE 4,15,0
              PRINT " Existent Pavement Deterioration Proportions File "
              FileSpec$ = " opened"
            CASE 3
              LOCATE 4,17,0
              PRINT "  View Pavement Deterioration Proportions File  "
              FileSpec$ = " seen"  :  Viewing$ = "Y"
            CASE 4
              Help$ = "C"
          END SELECT
          IF Help$ <> "C" THEN
            COLOR ColArr%(15),ColArr%(1)
            LOCATE 6,13,0
            PRINT "Enter name of proportions file to be" + FileSpec$
            'check the existence of the file
            FileExt$ = ".SHR"
            CALL FiNaCheck(FileExt$,FileChoice%)
            IF NewName$ <> "Y" THEN
              CLOSE # 2
              IF FileChoice% = 3 THEN
                CALL ViewShare(FileShare$)
                CLOSE # 2
              ELSE
                CALL GetShares(FileDat$,FileLra$,FileShare$)
              END IF
            END IF
          END IF
        LOOP UNTIL Help$ = "C"
      END IF
    END IF
    IF INSTR(1,"QC",Help$) <> 0 THEN EXIT LOOP
  LOOP UNTIL NewName$ <> "Y"
END SUB
'-------------------------------------------------------------------------
'compute share of load and non-load related factors on pavement deterioration
SUB GetShares(FileDat$,FileLra$,FileShare$)
      SHARED district$,county$,beginMP$,endMP$,region$,PBAindex$,ColArr%()
      SHARED Help$,IDhighway$,DistrictEnd$,CountyEnd$,MileBeg$,MileEnd$
      SHARED RegionEnd$,ShareLoad$,ShareNonLoad$,TheoryPBA$,ZeroPBA$,PBAdim$
```

```
SHARED TotalPBA$,PBAunit$,OnHelp$
'create the screen to display the results
CALL Frame
COLOR ColArr%(12),ColArr%(15)
LOCATE 24,31,0 : PRINT " ENTER : continue ";
COLOR ColArr%(15),ColArr%(1)
LOCATE 3,11,0
PRINT "Effect of Environmental Factors on Pavement Deterioration"
LOCATE 5,27,0 : PRINT "Performance Based Approach"
LOCATE 6,22,0 : PRINT "Oregon Department of Transportation"
LOCATE 7,32,0 : PRINT "Highway Division"
'get the highway ID
DO
  LOCATE 9,20,0 : PRINT "Enter the Highway ID:"
  COLOR ColArr%(12),ColArr%(15)
  LOCATE 9,43,0 : PRINT SPACE$(15)
  CALL InputString(StringIN$,9,44,13)
  IF Help$ <> "Q" THEN
    CALL Message
    LookHighway$ = StringIN$
    'get the records for LookHighway$ from .DAT file
    CALL LookDAT (LookHighway$,FileDat$,Help$)
    IF Help$ = "S" THEN
      'get the records for LookHighway$ from .LRA file
      CALL LookLra(LookHighway$,FileLra$,Help$)
      IF Help$ = "S" THEN
        'compute the different proportions
        ZeroPBA! = CSNG(CVD(TotalPBA$))
        TheoryPBA! = CVS(PBAindex$)
        a! = TheoryPBA!/ZeroPBA!
        d! = 1 - (SQR(1- ((1-a!)^2)))
        b! = ((a! * d!) * (1 + a! - d!))/(1 - a! - d!)
        c! = ((a! * d!) + (b! * d!))/(1-d!)
        'total shares
        ShareLoad! = a! + b!
        ShareNonLoad! = c! + d!
        'save the results
        CALL OpenShare(FileShare$)
        LSET IDhighway$ = LookHighway$
        LSET DistrictEnd$ = district$
        LSET CountyEnd$ = county$
        LSET MileBeg$ = beginMP$
        LSET MileEnd$ = endMP$
        LSET RegionEnd$ = region$
        LSET ShareLoad$ = MKS$(ShareLoad!)
        LSET ShareNonLoad$ = MKS$(ShareNonLoad!)
        LSET TheoryPBA$ = MKS$(TheoryPBA!)
        LSET ZeroPBA$ = MKS$(ZeroPBA!)
        LSET PBAdim$ = PBAunit$
        RecordNo& = (LOF(2)\118) + 1
        PUT # 2, RecordNo&
        CLOSE # 2
```

```
            'display the results
            COLOR ColArr%(15),ColArr%(1)
            FOR I%  = 9 TO 20
              LOCATE I%,10,0
              PRINT SPACE$(60)
            NEXT I%
            LOCATE 9,30,0  : PRINT "Highway Section:"
            LOCATE 10,30,0 : PRINT "District:"
            LOCATE 11,30,0 : PRINT "County:"
            LOCATE 12,30,0 : PRINT "Beginning Mile Post:"
            LOCATE 13,30,0 : PRINT "Ending Mile Post:"
            LOCATE 14,30,0 : PRINT "Region:"
            LOCATE 16,15,0 : PRINT "Pavement Deterioration Proportions:"
            LOCATE 17,15,0 : PRINT "Share of Load Related Factors:"
            LOCATE 18,15,0 : PRINT "Share of Non-Load Related Factors:"
            LOCATE 20,15,0 : PRINT "Theoretical PBA Index:"
            LOCATE 21,15,0 : PRINT "Zero Maintenance PBA Index:"
            COLOR ColArr%(12),ColArr%(15)
            LOCATE 9,51,0  : PRINT USING " \          \ "; LookHighway$
            LOCATE 10,51,0 : PRINT USING " \          \ "; DistrictEnd$
            LOCATE 11,51,0 : PRINT USING " \          \ "; CountyEnd$
            LOCATE 12,51,0 : PRINT USING " ####.####    "; CVS(beginMP$)
            LOCATE 13,51,0 : PRINT USING " ####.####    "; CVS(endMP$)
            LOCATE 14,51,0 : PRINT USING " \          \ "; region$
            LOCATE 17,50,0 : PRINT USING " ##.#### "; ShareLoad!
            LOCATE 18,50,0 : PRINT USING " ##.#### "; ShareNonLoad!
            LOCATE 20,43,0
            PRINT USING " #########.###### \            \ ";TheoryPBA!,PBAunit$
            LOCATE 21,43,0
            PRINT USING " #########.###### \            \ ";ZeroPBA!,PBAunit$
            DO : LOOP UNTIL INSTAT
            Quit$ = INKEY$
          END IF
        END IF
      END IF
    LOOP UNTIL INSTR(1,"QS",Help$) <> 0
END SUB
'.............................................................................
'display a wait message
SUB Message
    SHARED ColArr%()
    COLOR ColArr%(14),ColArr%(1)
    LOCATE 17,10,0 : PRINT CHR$(201) + STRING$(58,205) + CHR$(187)
    LOCATE 18,10,0 : PRINT CHR$(186) + SPACE$(58) + CHR$(186)
    LOCATE 19,10,0 : PRINT CHR$(186) + SPACE$(58) + CHR$(186)
    LOCATE 20,10,0 : PRINT CHR$(200) + STRING$(58,205) + CHR$(188)
    LOCATE 18,24,0 : PRINT "Retrieving Highway Section Data"
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 19,37,0 : PRINT " WAIT "
END SUB
'.............................................................................
'retrieve the record for LookHighway$ from the .DAT file
```

```
SUB LookDAT(LookHighway$,FileDat$,Help$)
       SHARED highwaynumber$,ColArr%()
       LOCAL Exists$
       'open the data file
       CALL OpenDat(FileDat$)
       EndFile& = LOF(1)\197
       Length% = LEN(LookHighway$)
       LookHighway$ = UCASE$(LookHighway$)
       FOR Record& = 1 TO EndFile&
         Exists$ = "Y"
         GET # 1, Record&
         IF LEFT$(UCASE$(highwaynumber$),Length%) = LookHighway$ THEN EXIT FOR
         Exists$ = "N"
       NEXT Record&
       CLOSE # 1
       IF Exists$ = "N" THEN
         CALL NoExists(LookHighway$,FileDat$)
         Help$ = "E"
       ELSE
         Help$ = "S"
       END IF
END SUB
'.............................................................................
'retrieve the record for LookHighway$ from the .LRA file
SUB LookLRA(LookHighway$,FileLra$,Help$)
       SHARED HighwayID$,ColArr%()
       LOCAL Exists$
       'open the data file
       CALL LRAopen(FileLra$)
       EndFile& = LOF(3)\590
       Length% = LEN(LookHighway$)
       LookHighway$ = UCASE$(LookHighway$)
       FOR Record& = 1 TO EndFile&
         Exists$ = "Y"
         GET # 3, Record&
         IF LEFT$(UCASE$(HighwayID$),Length%) = LookHighway$ THEN EXIT FOR
         Exists$ = "N"
       NEXT Record&
       CLOSE # 3
       IF Exists$ = "N" THEN
         CALL NoExists(LookHighway$,FileLra$)
         Help$ = "E"
       ELSE
         Help$ = "S"
       END IF
END SUB
'.............................................................................
'define the fields
SUB OpenDAT(FileDat$)
      SHARED highwaynumber$,district$,county$,beginMP$,endMP$,PBAindex$,region$
      OPEN FileDat$ FOR RANDOM AS # 1 LEN = 197
      FIELD #1, 25 AS highwaynumber$, 25 AS district$, 25 AS county$,_
```

```
                    4 AS beginMP$, 4 AS endMP$, 25 AS sectionEA$, 25 AS easj$,_
                    4 AS year$, 4 AS region$, 8 AS modulus$, 4 AS deviate$,_
                    4 AS deviation$, 4 AS StructuralNumber$, 4 AS PBAindex$,_
                    4 AS DynamicLoad$, 4 AS PlateRadius$, 4 AS PavementThickness$,_
                    4 AS SubgradePoisson$, 4 AS FieldSN$, 4 AS DeflectionFactor$,_
                    4 AS DeflSN$, 4 AS DeflDesign$
END SUB
'.......................................................................
'define the record for the share file
SUB OpenShare(FileShare$)
       SHARED IDhighway$,DistrictEnd$,CountyEnd$,MileBeg$,MileEnd$,RegionEnd$
       SHARED ShareLoad$,ShareNonLoad$,TheoryPBA$,ZeroPBA$,PBAdim$
       OPEN FileShare$ FOR RANDOM AS # 2 LEN = 118
       FIELD # 2, 25 AS IDhighway$, 25 AS DistrictEnd$, 25 AS CountyEnd$,_
                    4 AS MileBeg$, 4 AS MileEnd$, 4 AS RegionEnd$, 4 AS ShareLoad$,_
                    4 AS ShareNonLoad$, 4 AS TheoryPBA$, 4 AS ZeroPBA$,_
                    15 AS PBAdim$
END SUB
'.......................................................................
'display a message if the highway section is not found
SUB NoExists(LookHighway$,FileName$)
       SHARED ColArr%()
       LOCAL Length%
       COLOR ColArr%(14),ColArr%(1)
       Length% = 41 + LEN(LookHighway$)
       Col% = (80 - Length%) \ 2
       LOCATE 18,Col%,0 : PRINT "Highway Section: ";
       COLOR ColArr%(12),ColArr%(15)
       PRINT " " + LookHighway$ + " ";
       COLOR ColArr%(14),ColArr%(1)
       PRINT " couldn't be found in:"
       Length% = LEN(FileName$)
       Col% = (80 - Length%) \ 2
       LOCATE 19,Col%,0
       PRINT USING "&"; FileName$
       LOCATE 20,27,0 : PRINT " Press any key to continue "
       DO : LOOP UNTIL INSTAT
       COLOR ColArr%(15),ColArr%(1)
       FOR I% = 17 TO 20
         LOCATE I%,10,0
         PRINT SPACE$(60)
       NEXT I%
       Quit$ = INKEY$
END SUB
'-----------------------------------------------------------------------
'view the contents of a proportions file
SUB ViewShare(FileShare$)
       SHARED Help$,IDhighway$,DistrictEnd$,CountyEnd$,MileBeg$,MileEnd$,ColArr%()
       SHARED RegionEnd$,ShareLoad$,ShareNonLoad$,TheoryPBA$,ZeroPBA$,PBAdim$
       LOCAL TheoryFmt$,ZeroFmt$
       CALL OpenShare(FileShare$)
       CALL EqnScr(FileShare$)
```

```
      RecNo& = LOF(2) \ 118
      TheoryFmt$ = "Theoretical PBA Index: #########.###### \             \"
      ZeroFmt$ = "Zero Maintenance PBA Index: #########.###### \             \"
      FOR ViewSHR& = 1 TO RecNo&
        GET # 2, ViewSHR&
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 3,27,0 : PRINT "Performance Based Approach"
        LOCATE 4,22,0 : PRINT "Oregon Department of Transportation"
        LOCATE 5,32,0 : PRINT "Highway Division"
        LOCATE 8,20,0 : PRINT USING "Record No.: ####"; ViewSHR&
        LOCATE 9,20,0
        PRINT USING "Highway Section:  \           \ "; IDhighway$
        LOCATE 10,20,0
        PRINT USING "District:  \           \ "; DistrictEnd$
        LOCATE 11,20,0
        PRINT USING "County:  \           \ "; CountyEnd$
        LOCATE 12,20,0
        PRINT USING "Beginning Mile Post:  ####.####"; CVS(MileBeg$)
        LOCATE 13,20,0
        PRINT USING "Ending Mile Post:  ####.####"; CVS(MileEnd$)
        LOCATE 14,20,0
        PRINT USING "Region:  \           \ "; RegionEnd$
        LOCATE 16,15,0
        PRINT "Pavement Deterioration Proportions:"
        LOCATE 17,15,0
       PRINT USING "Share of Load Related Factors: ###.#### "; CVS(ShareLoad$)
       LOCATE 18,15,0
       PRINT USING "Share of Non Load Related Factors: ###.#### "; CVS(ShareNonLoad$)
        LOCATE 20,15,0
        PRINT USING TheoryFmt$; CVS(TheoryPBA$),PBAdim$
        LOCATE 21,15,0
        PRINT USING ZeroFmt$; CVS(ZeroPBA$),PBAdim$
        DO : LOOP UNTIL INSTAT
        Quit$ = INKEY$
        CALL ClScr(FileShare$)
        IF Quit$ = CHR$(27) THEN EXIT FOR
      NEXT ViewSHR&
END SUB
'---------------------------------------------------------------------------
```

Appendix A.3
PBAFFILE.BAS

```
'-------------------------------------------------------------------------
'
'                              PBAFFILE.BAS
'
'the following program was coded using Turbo BASIC version 1.1
'this file contains the subroutines used during the execution of the main file
'PBAFIELD.BAS to enter the name of the data file to use.
'-------------------------------------------------------------------------
'turn key trapping off
      $EVENT OFF
'-------------------------------------------------------------------------
SUB FiNaCheck(FileExt$,FileChoice%)
      SHARED ColArr%(),Editing$,OnHelp$,FileName$,FileEqn$,FileErr$,Viewing$
      SHARED FileLra$,Help$,FileShare$,FileDat$,NewName$
      LOCATE 7,13,0 : PRINT "You can add a disk drive address (eg. A:), or a"
      LOCATE 8,13,0 : PRINT "directory path (eg. C:\PBA\ ). DO NOT ADD A FILE"
      LOCATE 9,13,0 : PRINT "EXTENSION .  (40 characters maximum) : "
      LOCATE 10,15,0 : PRINT SPACE$(50)
17200 LOCATE 10,15,0
      COLOR ColArr%(12),ColArr%(15)
      PRINT SPACE$(41)
      Editing$ = "F"
      CALL InputString(StringIN$,10,16,40)
      IF Help$ <> "Q" THEN
        FileName$ = UCASE$(StringIN$)
        LOCATE 10,16,0
        Drive% = 0  ' to check free memory on disk
        'detect disk drive being used
        CharacterCheck$ = MID$(FileName$,2,1)
        LetterCheck% = INSTR(1,":\",CharacterCheck$)
        IF LetterCheck% <> 0 THEN
          Drive$ = MID$(FileName$,1,1)
          IF Drive$ = "A" THEN Drive% = 1
          IF Drive$ = "B" THEN Drive% = 2
          IF Drive$ = "C" THEN Drive% = 3
        END IF
        'find out which routine it is coming from
        SELECT CASE FileExt$
          CASE ".FEQ"
            FileEqn$ = FileName$ + ".FEQ"
            FileErr$ = FileEqn$
17300       OPEN FileEqn$ FOR RANDOM AS # 4 LEN = 472
            'determine if at least 12K bytes is available in the default
            'or indicated disk
            IF Help$ = "F" THEN EXIT SELECT
            NofRecords& = INT(FNbytes(Drive%) / 472) - 15
17305       IF NofRecords& < 15 THEN ERROR 61
            Test& = LOF(4)
          CASE ".LRA"
            FileLra$ = FileName$ + ".LRA"
            FileErr$ = FileLra$
17310       OPEN FileLra$ FOR RANDOM AS # 3 LEN = 605
```

```
        'determine if at least 12K bytes is available in the default
        'or indicated disk
        IF Help$ = "F" THEN EXIT SELECT
        NofRecords& = INT(FNbytes(Drive%) / 605) - 15
17315   IF NofRecords& < 15 THEN ERROR 61
        Test& = LOF(3)
      CASE ".DAT"
        FileDat$ = FileName$ + ".DAT"
        FileErr$ = FileDat$
17320   OPEN FileDat$ FOR RANDOM AS # 1 LEN = 197
        'determine if at least 12K bytes is available in the default
        'or indicated disk
        IF Help$ = "F" THEN EXIT SELECT
        NofRecords& = INT(FNbytes(Drive%) / 197) - 15
17325   IF NofRecords& < 15 THEN ERROR 61
        Test& = LOF(1)
      CASE ".SHR"
        FileShare$ = FileName$ + ".SHR"
        FileErr$ = FileShare$
17330   OPEN FileShare$ FOR RANDOM AS # 2 LEN = 197
        'determine if at least 12K bytes is available in the default
        'or indicated disk
        IF Help$ = "F" THEN EXIT SELECT
        NofRecords& = INT(FNbytes(Drive%) / 197) - 15
17335   IF NofRecords& < 15 THEN ERROR 61
        Test& = LOF(2)
    END SELECT
    'check the existence of the file
    IF Help$ = "F" THEN
      NewName$ = "Y"
    ELSE
      IF Test& = 0 THEN
        IF FileChoice% = 2 OR FileChoice% = 3 THEN
          CALL CloseFile(FileExt$)
          KILL FileErr$
          NewName$ = "Y"
17350     ERROR 33
        END IF
      END IF
      COLOR ColArr%(15),ColArr%(1)
      'if the length of FileDat$ is greater than 0 then it is assumed
      'that the file has been opened before, it already exists
      IF Test& > 0 AND Viewing$ = "N" THEN
        FOR ClearScr% = 13 TO 18
          LOCATE ClearScr%,11,0
          PRINT SPACE$(58)
        NEXT ClearScr%
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 14,12,0
        PRINT " " + FileErr$ + " ";
        COLOR ColArr%(15),ColArr%(1)
        PRINT " already exists."
```

```
LOCATE 16,12,0
PRINT "Do you want to overwrite/delete it (y/n) ?: "
LOCATE 17,12,0
PRINT "Do you want to add new information (y/n) ?: "
COLOR ColArr%(12),ColArr%(15)
DO
  CALL MenuChoice(Choice$,16,57)
  OverWrite$ = Choice$
  SELECT CASE OverWrite$
    CASE "N","n"  'if the user does not want to overwrite FileName$
      DO
        CALL MenuChoice(Choice$,17,57)
        add$ = Choice$
        SELECT CASE add$
          CASE "N","n"
            CALL CloseFile(FileExt$)
            CALL CleanScr
            NewName$ = "Y"
            EXIT LOOP
          CASE "Y","y"  ' if the user is to add data
            EXIT LOOP
        END SELECT
      LOOP
      EXIT IF
    CASE "Y","y"
      CALL CloseFile(FileExt$)
      COLOR ColArr%(15),ColArr%(1)
      FOR warn% = 13 TO 18 STEP 1  ' display the warning screen
        LOCATE warn%,11,0  :  PRINT SPACE$(56)
      NEXT warn%
      LOCATE 14,12,0 :  PRINT "The data stored in ";
      COLOR ColArr%(12),ColArr%(15)
      PRINT " " + FileErr$ + " "
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 15,12,0 : PRINT "are to be deleted"
      LOCATE 16,12,0 : PRINT "Enter 'y' to continue, 'n' to cancel"
      COLOR ColArr%(12),ColArr%(15)
      DO
        CALL MenuChoice(Choice$,16,52)
        warned$ = Choice$
        SELECT CASE warned$
          CASE "Y","y"           'delete the working file
            LOCATE 16,51,0
            KILL FileErr$
            EXIT LOOP
          CASE "N","n"
            COLOR ColArr%(15),ColArr%(1)
            NewName$ = "Y"
            EXIT LOOP
        END SELECT
      LOOP
      EXIT IF
```

```
              END SELECT
            LOOP
          END IF
        END IF
      ELSE
        NewName$ = "Y"
      END IF
END SUB
'.......................................................................
'close the opened file if needed
SUB CloseFile(FileExt$)
    SELECT CASE CloseFile$
      CASE ".FEQ"
        CLOSE # 4
      CASE ".LRA"
        CLOSE # 3
      CASE ".DAT"
        CLOSE # 1
      CASE ".SHR"
        CLOSE # 2
    END SELECT
END SUB
'.......................................................................
'the following function determines the amount of free space available in the
'default or indicated disk drive by invoking DOS Service 36H through DOS
'Interrupt 21H
DEF FNbytes(Drive%)
    REG 4,Drive%  'pass drive code to DL:DX (0=default,1=A:,2=B:,3=C:,etc.)
    REG 1,&H3600  ' pass function code to AH:AX
    CALL INTERRUPT &H21
    FNbytes = CSNG(REG(2))*REG(3)*REG(1) 'determine free space (AX*BX*CX)
END DEF
'-----------------------------------------------------------------------
SUB FileScreen
      SHARED ColArr%()
18250 COLOR ColArr%(15),ColArr%(1)
      CLS
      LOCATE 3,10,0  :  PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      LOCATE 4,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      LOCATE 5,10,0  :  PRINT CHR$(195) + STRING$(58,196) + CHR$(180)
      FOR Fb% = 6 TO 10
        LOCATE Fb%,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      NEXT Fb%
      LOCATE 11,10,0 :  PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
      LOCATE 12,10,0 :  PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      FOR Fb% = 13 TO 18
        LOCATE Fb%,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      NEXT Fb%
      LOCATE 19,10,0 :  PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
END SUB
'-----------------------------------------------------------------------
SUB CleanScr
```

```
      SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 4,11,0  :  PRINT SPACE$(58)
      FOR Fb% = 6 TO 10
        LOCATE Fb%,11,0
        PRINT SPACE$(58)
      NEXT Fb%
      FOR Fb% = 13 TO 18
        LOCATE Fb%,11,0
        PRINT SPACE$(58)
      NEXT Fb%
      FOR Fb% = 21 TO 23
        LOCATE Fb%,11,0
        PRINT SPACE$(58)
      NEXT Fb%
      LOCATE 24,11,0 :  PRINT SPACE$(58);
END SUB
'---------------------------------------------------------------------------
'the following subroutine is used to enter different types of data
SUB InputData(InDat$)
      SHARED ColArr%(),HighwayNo$,SubEqId$,Neck$,NoVar%,Editing$,Help$
      SHARED VarNameX$,UnitsX$,VarX$,Col%,NoData%,Box%,Dat!(),MeasUnits$
      SHARED ESALu!(),PaveID$(),MinNo%,MaxNo%,ColIn%,NoBox%,VarId$,MeasName$
      SHARED OnHelp$,DatBck$
      CALL Frame
      DatBck$ = InDat$
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 24,16,0  :  PRINT " F1 : Help ";
      LOCATE 24,28,0  :  PRINT " F4 : Edit ";
      LOCATE 24,40,0  :  PRINT " ESC : Quit ";
      LOCATE 24,53,0  :  PRINT " F9 : End ";
      COLOR ColArr%(15),ColArr%(1)
      SELECT CASE InDat$
        CASE "E"    'when on UserEqn routine
          LOCATE 3,20,0 : PRINT USING "Pavement ID : & "; LEFT$(HighwayNo$,10)
          LOCATE 4,20,0 : PRINT USING "Equation ID : & "; LEFT$(SubEqId$,10)
          LOCATE 5,20,0 : PRINT USING "Variable    : & "; LEFT$(Neck$,25)
          LOCATE 6,20,0 : PRINT USING "No. Variable: # "; NoVar%
          CALL DataFrame("Y",39,20,19)
          COLOR ColArr%(12),ColArr%(15)
          LOCATE 9,25,0
          PRINT USING " Values for variable & "; SubEqId$ + VarId$
          LOCATE 10,26,0 : PRINT " YEAR "
          LOCATE 10,45,0 : PRINT " MEASURE "
        CASE "T"    'when on UserEqn routine
          LOCATE 5,20,0  :  PRINT "Enter the variable name and units:"
          LOCATE 6,25,0  :  PRINT "Pavement Section ID:"
          LOCATE 7,25,0  :  PRINT "Variable Name:"
          LOCATE 8,25,0  :  PRINT "Variable Units:"
          LOCATE 9,25,0  :  PRINT "Variable ID:"
          COLOR ColArr%(12),ColArr%(15)
          LOCATE 3,30,0  :  PRINT " FIELD MEASUREMENTS "
```

```
  LOCATE 4,14,0
  PRINT " Independent Variable of Field Performance Equation "
  LOCATE 6,46,0  :  PRINT SPACE$(12)
  LOCATE 7,41,0  :  PRINT SPACE$(30)
  LOCATE 8,41,0  :  PRINT SPACE$(30)
  LOCATE 9,41,0  :  PRINT SPACE$(7)
  CALL InputString(StringIN$,6,47,10)
  IF Help$ <> "Q" THEN
    HighwayNo$ = StringIN$
    CALL InputString(StringIN$,7,42,25)
    IF Help$ <> "Q" THEN
      VarNameX$ = StringIN$
      CALL InputString(StringIN$,8,42,25)
      IF Help$ <> "Q" THEN
        UnitsX$ = StringIN$
        CALL InputString(StringIN$,9,42,5)
        IF Help$ <> "Q" THEN
          VarX$ = StringIN$
          COLOR ColArr%(15),ColArr%(1)
          FOR I% = 5 TO 9
            LOCATE I%,20,0
            PRINT SPACE$(52)
          NEXT I%
        END IF
      END IF
      CALL DataFrame("N",39,20,19)
      COLOR ColArr%(12),ColArr%(15)
      Col% = LEN(HighwayNo$) + 15
      Col% = (80 - Col%) / 2
      LOCATE 5,Col%,0
      PRINT USING " Pavement ID : & "; HighwayNo$
      LOCATE 10,28,0 :  PRINT "    Year         " + VarX$ + "      "
    END IF
  END IF
CASE "Z"      'get the pavement sections
  LOCATE 5,20,0 : PRINT "Enter the Highway Section ID: "
  COLOR ColArr%(12),ColArr%(15)
  LOCATE 3,34,0  :  PRINT " PBA Index "
  LOCATE 4,22,0  :  PRINT " Zero Level of Routine Maintenance "
  LOCATE 5,50,0  :  PRINT SPACE$(12)
  CALL InputString(StringIN$,5,51,10)
  IF Help$ <> "Q" THEN
    HighwayNo$ = StringIN$
    COLOR ColArr%(15),ColArr%(1)
    LOCATE 5,20,0  :  PRINT SPACE$(45)
    Col% = LEN(HighwayNo$) + 15
    Col% = (80 - Col%)\2
    CALL DataFrame("N",39,20,19)
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 5,Col%,0
    PRINT USING " Highway ID: & "; HighwayNo$
    LOCATE 10,29,0 : PRINT " Pavement Sections ID "
```

```
    END IF
   CASE "M"    'get measure of zero level of routine maintenance
    LOCATE 5,20,0  :  PRINT "Enter the measure name and units:"
    LOCATE 6,25,0  :  PRINT "Measure Name:"
    LOCATE 7,25,0  :  PRINT "Measure Units:"
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 3,34,0  :  PRINT " PBA Index "
    LOCATE 4,22,0  :  PRINT " Zero Level of Routine Maintenance "
    LOCATE 6,40,0  :  PRINT SPACE$(15)
    LOCATE 7,40,0  :  PRINT SPACE$(15)
    CALL InputString(StringIN$,6,41,13)
    IF Help$ <> "Q" THEN
      MeasName$ = StringIN$
      CALL InputString(StringIN$,7,41,13)
      IF Help$ <> "Q" THEN
        MeasUnits$ = StringIN$
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 5,20,0 : PRINT SPACE$(35)
        CALL DataFrame("Y",61,9,30)
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 5,Col%,0
        PRINT USING " Highway ID: & "; HighwayNo$
        LOCATE 9,20,0
        PRINT " Measure of Level of Routine Maintenance "
        LOCATE 10,13,0 : PRINT " Section "
        LOCATE 10,28,0 : PRINT " Measure "
        LOCATE 10,44,0 : PRINT " Section "
        LOCATE 10,57,0 : PRINT " Measure "
      END IF
    END IF
END SELECT
IF Help$ <> "Q" THEN
  IF INSTR(1,"ETZ",InDat$) <> 0 THEN
    COLOR ColArr%(15),ColArr%(1)
    IF InDat$ = "Z" THEN
      LOCATE 7,10,0
      PRINT "Input the number of Pavement Sections "
      MaxNo% = 20 : MinNo% = 2
    ELSEIF InDat$ = "E" THEN
      LOCATE 7,20,0
      PRINT "Number of measurements (max: 10):"
      MaxNo% = 10 : MinNo% = 1
    ELSEIF InDat$ = "T" THEN
      LOCATE 7,20,0
      PRINT "Input the number of measurements"
      MaxNo% = 10 : MinNo% = 1
    END IF
    DO
      IF InDat$ = "Z" THEN
        LOCATE 8,10,0
        PRINT "(minimum: 2  maximum: 20 sections) :"
      ELSEIF InDat$ = "E" THEN
```

```
            LOCATE 7,43,0
            PRINT "(max: 10)"
          ELSEIF InDat$ = "T" THEN
            LOCATE 8,20,0
            PRINT "(maximum of 10 measurements) :"
          END IF
          COLOR ColArr%(12),ColArr%(15)
          IF INSTR(1,"ZT",InDat$) <> 0 THEN
18050       LOCATE 8,51,0 : PRINT SPACE$(4)
            CALL InputInteger(Value%,8,52,3)
          ELSEIF InDat$ = "E" THEN
            LOCATE 7,54,0 : PRINT SPACE$(4)
            CALL InputInteger(Value%,7,55,3)
          END IF
          IF Help$ = "Q" THEN EXIT LOOP
          NoData% = Value%
          'checking that the number of measurements is within range
          IF NoData% < MinNo% OR NoData% > MaxNo% THEN
            COLOR ColArr%(18),ColArr%(1)
          ELSE
            EXIT LOOP
          END IF
        LOOP
      END IF
    END IF
    IF Help$ <> "Q" THEN
      COLOR ColArr%(15),ColArr%(1)
      Editing$ = "M"
      IF InDat$ = "Z" THEN
        LOCATE 8,10,0
        PRINT "(minimum: 2  maximum: 20 sections) :"
        NoBox% = NoData%
        DIM DYNAMIC PaveID$(1:10,1:2)
      ELSEIF InDat$ = "M" THEN
        NoBox% = NoData%
        DIM DYNAMIC Dat!(1:10,1:2)
      ELSEIF InDat$ = "T" THEN
        LOCATE 8,20,0
        PRINT "(maximum of 10 measurements) :"
        NoBox% = NoData% * 2
        DIM DYNAMIC Dat!(1:10,1:2)
      ELSEIF InDat$ = "E" THEN
        LOCATE 7,43,0
        PRINT "(max: 10)"
        NoBox% = NoData% * 2
        DIM DYNAMIC Dat!(1:10,1:2)
      END IF
      'the following loop enters the data
      Cbck% = ColArr%(1)   :  BckC% = ColArr%(10)
      DataIn$ = "N"  :  InCell$ = "Y"
      DO
        'position of labels and cursor
```

```
IF InDat$ = "M" THEN
  None% = 41  :  Ntwo% = 10
  Cone% = 55  :  Ctwo% = 24
  Lbl0% = 2   :  LblT% = 1
ELSE
  None% = 41  :  Ntwo% = 21
  Cone% = 45  :  Ctwo% = 25
END IF
ColIn% = 2  :  ColOut% = 1
Count% = 1  :  Brow% = 12
Box% = 1  :  Pointer% = 1
WHILE Pointer% <= NoBox%
  COLOR ColArr%(15),Cbck%
  SWAP None%,Ntwo%
  'print the data number
  IF InDat$ = "Z" AND None% = 41 THEN
    LOCATE Brow%,None%,0
    PRINT USING " ## "; Count%
    INCR Count%
  ELSEIF INSTR(1,"ET",InDat$) <> 0 AND None% = 41 THEN
    LOCATE Brow%,None%,0
    PRINT USING " ## "; (Count%-1)
  ELSEIF InDat$ = "M" THEN
    SWAP Lbl0%,LblT%
    LOCATE Brow%,None%,0
    PRINT USING " \          \ "; PaveID$(Brow%-11,Lbl0%)
  ELSEIF None% = 21 THEN
    LOCATE Brow%,None%,0
    PRINT USING " ## "; Count%
    INCR Count%
  END IF
  'display the input cell
  COLOR ColArr%(15),ColArr%(10)
  SWAP Cone%,Ctwo%
  LOCATE Brow%,Cone%,0
  PRINT SPACE$(15)
  'check if on loop to enter data
  IF DataIn$ = "Y" THEN
    SWAP ColIn%,ColOut%
    IF InDat$ = "Z" THEN
      'enter pavement sections ID
      CALL InputString(StringIN$,Brow%,Cone%+1,10)
      IF Help$ = "Q" THEN
        ERASE PaveID$
      ELSE
        PaveID$(Brow%-11,ColIn%) = StringIN$
        COLOR ColArr%(15),ColArr%(1)
        LOCATE Brow%,None%,0
        PRINT USING " ## "; (Count%-1)
      END IF
    ELSE
      'enter data for independent variable, or subequation variable
```

```
                CALL InputReal(Value!,Brow%,Cone%+1,11)
                IF Help$ = "Q" THEN
                  ERASE Dat!
                ELSE
                  Dat!(Brow%-11,ColIn%) = Value!
                  LOCATE Brow%,Cone%,0
                  PRINT USING " ###########    "; Dat!(Brow%-11,ColIn%)
                  COLOR ColArr%(15),ColArr%(1)
                  LOCATE Brow%,None%,0
                  'display the label
                  IF InDat$ = "M" THEN
                    PRINT USING " \          \ "; PaveID$(Brow%-11,ColIn%)
                  ELSE
                    PRINT USING " ## "; (Count%-1)
                  END IF
                END IF
              END IF
            END IF
            IF Cone% = 45 OR Cone% = 55 THEN
              INCR Brow%
              INCR Box%
            END IF
            INCR Pointer%
            IF Help$ = "Q" THEN EXIT LOOP
          WEND
          SWAP Cbck%,BckC%
          SWAP DataIn$,InCell$
        LOOP UNTIL DataIn$ = "N"
        IF Help$ <> "Q" THEN
          SELECT CASE InDat$
            CASE "T"
              DIM DYNAMIC ESALu!(0:10)
              FOR I% = 1 TO 10
                ESALu!(I%) = Dat!(I%,2)
              NEXT I%
              ERASE Dat!
            CASE ELSE
              Editing$ = "E"  'flag used for UserEquation routine
          END SELECT
        END IF
      END IF
END SUB
'..........................................................................
'create the frame for inputing data
SUB DataFrame(UpBound$,Bound%,Edge%,Cln%)
      SHARED ColArr%()
      COLOR ColArr%(14),ColArr%(1)
      IF UpBound$ = "Y" THEN
        'when from the ZeroLevel Routine
        LOCATE 8,Edge%,0
        PRINT CHR$(218) + STRING$(Bound%,196) + CHR$(191)
        LOCATE 9,Edge%,0
```

```
      PRINT CHR$(179) + SPACE$(Bound%) + CHR$(179)
    ELSE
      'when from the UserEqn, or ZeroLevel routines
      LOCATE 9,Edge%,0
      PRINT CHR$(218) + STRING$(Bound%,196) + CHR$(191)
    END IF
    LOCATE 10,Edge%,0 : PRINT CHR$(179) + SPACE$(Bound%) + CHR$(179)
    LOCATE 11,Edge%,0
    PRINT CHR$(195)+STRING$(Cln%,196)+CHR$(194)+STRING$(Cln%,196)+CHR$(180)
    FOR I% = 12 TO 21
      LOCATE I%,Edge%,0
      PRINT CHR$(179) + SPACE$(Cln%) + CHR$(179) + SPACE$(Cln%) + CHR$(179)
    NEXT I%
    LOCATE 22,Edge%,0
    PRINT CHR$(192)+STRING$(Cln%,196)+CHR$(193)+STRING$(Cln%,196)+CHR$(217)
END SUB
'-----------------------------------------------------------------------------
'create the data collection screen for the identification data
SUB Frame
    SHARED ColArr%()
    COLOR ColArr%(15),ColArr%(1)
    CLS
    LOCATE 1,2,0  :  PRINT CHR$(201) + STRING$(74,205) + CHR$(187)
    FOR I% = 2 TO 22
      LOCATE I%,2,0
      PRINT CHR$(186) + SPACE$(74) + CHR$(186)
    NEXT I%
    LOCATE 23,2,0 :  PRINT CHR$(204) + STRING$(74,205) + CHR$(185)
    LOCATE 24,2,0 :  PRINT CHR$(186) + SPACE$(74) + CHR$(186);
    LOCATE 25,2,0 :  PRINT CHR$(200) + STRING$(74,205) + CHR$(188);
END SUB
'-----------------------------------------------------------------------------
'routine used to enter an integer value
SUB InputInteger(Value%,RowDat%,ColDat%,MaxNoChr%)
    SHARED Help$
    Value% = 0
    CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
    IF Help$ <> "Q" THEN Value% = CINT(VAL(StringIN$))
END SUB
'-----------------------------------------------------------------------------
'routine used to enter a long integer value
SUB InputLong(Value&,RowDat%,ColDat%,MaxNoChr%)
    SHARED Help$
    Value& = 0
    CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
    IF Help$ <> "Q" THEN Value& = CLNG(VAL(StringIN$))
END SUB
'-----------------------------------------------------------------------------
'routine used to enter a real value
SUB InputReal(Value!,RowDat%,ColDat%,MaxNoChr%)
    SHARED Help$
    Value! = 0
```

```
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
      IF Help$ <> "Q" THEN Value! = CSNG(VAL(StringIN$))
END SUB
'-----------------------------------------------------------------------
'routine used to enter a string
SUB InputDouble(Value#,RowDat%,ColDat%,MaxNoChr%)
      SHARED Help$
      Value# = 0
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
      IF Help$ <> "Q" THEN Value# = CDBL(VAL(StringIN$))
END SUB
'-----------------------------------------------------------------------
'routine used to enter data
SUB InputString(StringIN$,RowDat%,ColDat%,MaxNoChr%)
      SHARED Help$
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
END SUB
'-----------------------------------------------------------------------
'this routine is used to select a menu option
SUB MenuChoice(Choice$,RowIn%,ColIn%)
      SHARED Help$,ColArr%(),OnHelp$
      LOCAL ExitSub$
      COLOR ColArr%(12),ColArr%(15)
      DO
        LOCATE RowIn%,ColIn%-1,0  :  PRINT SPACE$(3);
        Help$ = "N"
        DO  :  LOCATE RowIn%,ColIn%,1  :  LOOP UNTIL INSTAT
        IF Help$ <> "Y" THEN
          Choice$ = INKEY$
          IF Choice$ = CHR$(27) THEN
            Help$ = "Q"
            ExitSub$ = CHR$(13)
          ELSEIF MID$(Choice$,2,1) = CHR$(59) THEN
            CALL PBAhelp(OnHelp$)
          ELSEIF Choice$ <> CHR$(13) OR Choice$ <> CHR$(8) THEN
            LOCATE RowIn%,ColIn%,0  :  PRINT Choice$
            LOCATE RowIn%,ColIn%+1,1
            DO :  LOOP UNTIL INSTAT
            ExitSub$ = INKEY$
          END IF
        END IF
      LOOP UNTIL ExitSub$ = CHR$(13)
END SUB
'.......................................................................
'this routine enters a string of up to MaxNoChr% characters
SUB EnterData(DataRow%,DataCol%,InString$,MaxNoChr%)
10900 SHARED Help$,Editing$,CarIn$,OnHelp$,EditEq$
      LOCAL LetPos%,NoChr%,Lines%,StartCol%
      InString$ = STRING$(MaxNoChr%,0) : LetPos% = 1
      StartCol% = DataCol%  :  Help$ = ""
      NoChr% = 1  :  DataCol% = StartCol%
      DO WHILE NoChr% <= MaxNoChr%
```

```
        Help$ = "N"
        DO
          LOCATE DataRow%,DataCol%,1
        LOOP UNTIL INSTAT
        Letter$ = INKEY$
        SELECT CASE Letter$
          CASE CHR$(27)
            Help$ = "Q"
          CASE CHR$(13)
            EXIT LOOP
          CASE CHR$(8)
            IF DataCol% > StartCol% THEN   'avoid backspacing
              DECR DataCol% : DECR NoChr%   :  DECR LetPos%
              LOCATE DataRow%,DataCol%,0   :  PRINT SPACE$(1)
              MID$(InString$,LetPos%,1) = CHR$(0)
            END IF
          CASE ELSE
            SELECT CASE MID$(Letter$,2,1)
              CASE CHR$(59)
                CALL PBAhelp(OnHelp$)
              CASE CHR$(60)
                IF Editing$ = "E" THEN
                  CALL NeEqE
                  Help$ = "E"
                END IF
              CASE CHR$(61)
                IF Editing$ = "Eq" THEN
                  Help$ = "Q"
                  EditEq$ = "Y"
                END IF
              CASE CHR$(62)
                IF Editing$ = "M" THEN CALL ManTrEd
              CASE ELSE
                IF LEN(Letter$) = 1 THEN
                  LOCATE DataRow%,DataCol%,0
                  PRINT Letter$
                  MID$(InString$,LetPos%,1) = Letter$
                  INCR DataCol% : INCR NoChr%   :  INCR LetPos%
                END IF
            END SELECT
      END SELECT
      IF INSTR(1,"QE",Help$) <> 0 THEN EXIT LOOP
    LOOP
    IF INSTR(1,"EQ",Help$) = 0 THEN
      InString$ = LEFT$(InString$,LetPos%-1)
    END IF
END SUB
'..............................................................................
'display the equation screen
SUB EqnScr(FileName$)
      SHARED ColArr%()
      COLOR ColArr%(14),ColArr%(1)
```

```
        CLS
        LOCATE 1,1,0 : PRINT CHR$(218) + STRING$(78,196) + CHR$(191)
        FOR Frame% = 2 TO 22
          LOCATE Frame%,1,0
          PRINT CHR$(179) + SPACE$(78) + CHR$(179)
        NEXT Frame%
        IF RIGHT$(FileName$,4) = ".LRA" THEN LOCATE 8,1,0 ELSE LOCATE 7,1,0
        PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
        IF RIGHT$(FileName$,4) = ".FEQ" THEN
          LOCATE 19,1,0
          PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
        END IF
        LOCATE 23,1,0 : PRINT CHR$(195) + STRING$(78,196) + CHR$(180);
        LOCATE 24,1,0 : PRINT CHR$(179) + SPACE$(78) + CHR$(179);
        LOCATE 25,1,0 : PRINT CHR$(192) + STRING$(78,196) + CHR$(217);
        COLOR ColArr%(12),ColArr%(15)
        Col% = ((80 - LEN(FileName$))/2) - 1
        LOCATE 1,Col%,0 : PRINT "  " + UCASE$(FileName$) + "  "
        LOCATE 24,24,0  : PRINT " ENTER : Continue ";
        LOCATE 24,43,0  : PRINT   " ESC : Quit ";
END SUB
'...............................................................................
'clean equation screen
SUB ClScr(FileName$)
        SHARED ColArr%()
        COLOR ColArr%(14),ColArr%(1)
        FOR Frame% = 2 TO 22
          LOCATE Frame%,2,0
          PRINT SPACE$(78)
        NEXT Frame%
        IF RIGHT$(FileName$,4) = ".LRA" THEN LOCATE 8,1,0 ELSE LOCATE 7,1,0
        PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
        IF RIGHT$(FileName$,4) = ".FEQ" THEN
          LOCATE 19,1,0
          PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
        END IF
END SUB
'-------------------------------------------------------------------------------
'determine the type of monitor being used: monochrome or color
SUB Monitor(VideoMode%)
        SHARED ColArr%()
        DIM DYNAMIC ColArr%(1:20)
        DEF SEG = 0
        VideoMode% = PEEK(&H449)
        SELECT CASE VideoMode%
          CASE 5,6,7,13    'black and white monitor
            ColArr%(1) = 0   : ColArr%(4) = 7   : ColArr%(5) = 16
            ColArr%(6) = 15  : ColArr%(10) = 7  : ColArr%(12) = 0
            ColArr%(13) = 7  : ColArr%(14) = 15 : ColArr%(15) = 15
            ColArr%(16) = 16 : ColArr%(17) = 7  : ColArr%(18) = 23
            ColArr%(19) = 0  : ColArr%(20) = 7
          CASE ELSE       'color monitor
```

```
        ColArr%(1) = 1   : ColArr%(4) = 4   : ColArr%(5) = 31
        ColArr%(6) = 10  : ColArr%(10) = 10 : ColArr%(12) = 4
        ColArr%(13) = 4  : ColArr%(14) = 14 : ColArr%(15) = 15
        ColArr%(20) = 20 : ColArr%(16) = 26 : ColArr%(18) = 28
        ColArr%(19) = 30 : ColArr%(20) = 31
    END SELECT
END SUB
'-----------------------------------------------------------------------------
```

Appendix A.4
PBAFEDIT.BAS

```
'---------------------------------------------------------------------------
'
'                               PBAFEDIT.BAS
'
'the following program was coded using Turbo BASIC version 1.1
'this file contains the subroutines used during the execution of the main file
'PBAFIELD.BAS to change data on the different input screens.
'---------------------------------------------------------------------------
'turn key trapping off
      $EVENT OFF
'---------------------------------------------------------------------------
'this subroutine edits values for the InputData subroutine
SUB ManTrEd
     SHARED DatBck$,Help$,Box%,NoData%,Dat!(),HighwayNo$,SubEqID$,Neck$
     SHARED ColArr%(),MTedit%,MtR%,MtC%,DaR%,DaC%,MtLaR%,ColH%,NoVar%
     SHARED PaveID$(),MinNo%,MaxNo%,ColIn%,NoBox%,Col%
     CALL EditScreen
     Help$ = "E"
     IF Help$ <> "Q" THEN
       COLOR ColArr%(15),ColArr%(1)
       LOCATE 24,16,0  :  PRINT SPACE$(50);
       MtLaR% = (Box%+11)
       IF DatBck$ = "M" THEN
         MtR% = 12
         MtC% = 24
       ELSEIF INSTR(1,"ZT",DatBck$) <> 0 THEN
         MtR% = 8       ' controlling variables
         MtC% = 51
       ELSEIF DatBck$ = "E" THEN
         MtR% = 7
         MtC% = 54
       END IF
       DaR% = 1   :  DaC% = 1   ' for the MoDo routine
       DO
8020     CALL PrTra(MtR%,DaC%,ColArr%(14),ColArr%(12))
         DO
8025       DO  :  LOOP UNTIL INSTAT
           MTedit$ = INKEY$   ' after pressing RETURN the data is to be changed
           SELECT CASE MTedit$
             CASE CHR$(13)
               'the following IF block changes the number of boxes if needed
               IF MtR% <= 8 AND DatBck$ <> "M" THEN
                 DO
8027               LOCATE MtR%,MtC%,0  :  PRINT SPACE$(4)
                   LOCATE MtR%,(MtC%+1),1
                   CALL EditInteger(EdValue%,MtR%,(MtC%+1),2)
                   NewNoM% = EdValue%
                   SELECT CASE NewNoM%
                     CASE MinNo% TO MaxNo%
                       'the following IF block increases the number of boxes
                       IF NewNoM% > NoData% THEN
                         ' change the old variables
```

```
    NewM% = NoBox% + 1
    IF DatBck$ = "Z" THEN
      NoBox% = NewNoM%
    ELSE
      NoBox% = NewNoM% * 2
    END IF
    EdCnt% = NoData% + 1
    'get the last row, and the columns
    EdOne% = 41 : EdTwo% = 21
    CedO% = 45  : CedT% = 25
    IF DatBck$ = "Z" THEN   'if on zero level
      IF (NewM% MOD 2) = 0 THEN
        SWAP EdOne%,EdTwo%
        SWAP CedO%,CedT%
      END IF
      'get the last row
      IF (NoData% MOD 2) = 0 THEN
        EdRow% = (NoData% / 2) + 12
      ELSE
        EdRow% = ((NoData%-1)/2) + 12
      END IF
    ELSE     'if on subequation or independent variable
      EdRow% = 12 + NoData%
    END IF
    ' display the new boxes
    WHILE NewM% <= NoBox%
      COLOR ColArr%(15),ColArr%(1)
      SWAP EdOne%,EdTwo%
      'print the data number
      IF DatBck$ = "Z" AND EdOne% = 41 THEN
        LOCATE EdRow%,EdOne%,0
        PRINT USING " ## "; EdCnt%
        INCR EdCnt%
      ELSEIF INSTR(1,"ET",DatBck$) <> 0 AND EdOne% = 41 THEN
        LOCATE EdRow%,EdOne%,0
        PRINT USING " ## "; (EdCnt%-1)
      ELSEIF EdOne% = 21 THEN
        LOCATE EdRow%,EdOne%,0
        PRINT USING " ## "; EdCnt%
        INCR EdCnt%
      END IF
      'display the input cell
      COLOR ColArr%(15),ColArr%(10)
      SWAP CedO%,CedT%
      LOCATE EdRow%,CedO%,0
      PRINT SPACE$(15)
      INCR NewM%
      IF EdOne% = 41 THEN INCR EdRow%
    WEND
    NoData% = NewNoM%
  END IF
  COLOR ColArr%(12),ColArr%(15)
```

```
                        LOCATE MtR%,MtC%,0 : PRINT USING " ## "; NoData%
                        EXIT LOOP
                    END SELECT
                  LOOP
                ELSE
                  LOCATE MtR%,MtC%,0  :  PRINT SPACE$(15)
                  IF DatBck$ = "Z" THEN
                    CALL EditString(EdStr$,MtR%,(MtC%+1),12)
                    PaveID$(DaR%,DaC%) = EdStr$
                  ELSE
                    CALL EditReal(EdValue!,MtR%,(MtC%+1),11)
                    Dat!(DaR%,DaC%) = EdValue!
                  END IF
                  EXIT LOOP
                END IF
              CASE CHR$(27)
                EXIT LOOP
              CASE ELSE
                SELECT CASE MID$(MTedit$,2,1)
                  CASE CHR$(72),CHR$(80)
                    CALL MoDo
                  CASE CHR$(75),CHR$(77)
                    CALL MoRi
                END SELECT
            END SELECT
          LOOP
        LOOP UNTIL MTedit$ = CHR$(27)
        CALL PrTra(MtR%,DaC%,ColArr%(15),ColArr%(10))
 8030   CALL ClEdSc
        SELECT CASE DatBck$
          CASE "E"
            LOCATE 3,20,0 : PRINT USING "Pavement ID : &"; LEFT$(HighwayNo$,12)
            LOCATE 4,20,0 : PRINT USING "Equation ID : &"; LEFT$(SubEqID$,10)
          CASE "T"
            COLOR ColArr%(12),ColArr%(15)
            LOCATE 3,30,0   : PRINT " FIELD MEASUREMENTS "
            LOCATE 5,Col%,0
            PRINT USING " Pavement ID: & "; LEFT$(HighwayNo$,12)
          CASE "Z","M"
            COLOR ColArr%(12),ColArr%(15)
            LOCATE 3,34,0 : PRINT " PBA Index "
            LOCATE 4,22,0 : PRINT " Zero Level of Routine Maintenance "
        END SELECT
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 24,16,0  :  PRINT " F1 : Help ";
        LOCATE 24,28,0  :  PRINT " F4 : Edit ";
        LOCATE 24,40,0  :  PRINT " ESC : Quit ";
        LOCATE 24,53,0  :  PRINT " F9 : End ";
        COLOR ColArr%(15),ColArr%(10)
8035 END IF
END SUB
'.......................................................................
```

```
SUB PrTra(MtR%,DaC%,Fore%,Back%)
     SHARED MtC%,Dat!(),NoData%,DaR%,PaveID$(),DatBck$
     LOCATE MtR%,MtC%,0   ' go to the uppermost left data box
     COLOR Fore%,Back%
     IF MtR% <= 8 THEN
       PRINT USING " ## "; NoData%
     ELSEIF DaC% = 1 THEN
       IF DatBck$ = "Z" THEN
         PRINT USING " \          \  "; PaveID$(DaR%,DaC%)
       ELSEIF DatBck$ = "M" THEN
         PRINT USING " ###########   "; Dat!(DaR%,DaC%)
       ELSE
         PRINT USING "  ####         "; Dat!(DaR%,DaC%)
       END IF
     ELSEIF DaC% = 2 THEN
       IF DatBck$ = "Z" THEN
         PRINT USING " \          \  "; PaveID$(DaR%,DaC%)
       ELSE
         PRINT USING " ###########   "; Dat!(DaR%,DaC%)
       END IF
     END IF
END SUB
'..........................................................................
'the MoDo subroutine moves the cursor up or down for editing
SUB MoDo
      SHARED ColArr%(),MtR%,MtC%,DaR%,DaC%,MtLaR%,ColIn%
      CALL PrTra(MtR%,DaC%,ColArr%(15),ColArr%(10))
      IF MtR% <= 8 THEN
        MtR% = 12  :  MtC% = 25
      ELSEIF DaC% = 1 OR DaC% = 2 THEN
        INCR MtR%  :  INCR DaR%
      END IF
      IF MtR% > MtLaR% THEN      'this moves the cursor to the
        MtR% = 12                'uppermost data box if the box where the
        DaR% = 1                 'cursor was located when this routine was
      END IF                     'called, is reached
      IF MtR% = MtLaR% AND DaC% >= ColIn% THEN
        MtR% = 12
        DaR% = 1
      END IF
8100  CALL PrTra(MtR%,DaC%,ColArr%(14),ColArr%(12))
END SUB
'..........................................................................
'the MoRi subroutine moves the cursor left or right
SUB MoRi
      SHARED ColArr%(),MtR%,MtC%,DaR%,DaC%,MtLaR%,DatBck$
      IF MtR% > 8 THEN
        CALL PrTra(MtR%,DaC%,ColArr%(15),ColArr%(10))
        INCR DaC%
        IF DatBck$ = "M" THEN INCR MtC%,31 ELSE INCR MtC%,20
        IF DaC% > 2 THEN
          DaC% = 1
```

```
          IF DatBck$ = "M" THEN MtC% = 24 ELSE MtC% = 25
        END IF
        IF MtR% = MtLaR% AND DaC% = 2 THEN
          DaC% = 1
          IF DatBck$ = "M" THEN MtC% = 24 ELSE MtC% = 25
        END IF
        CALL PrTra(MtR%,DaC%,ColArr%(14),ColArr%(12))
      END IF
END SUB
'------------------------------------------------------------------------
'to make changes during NewEquation routine
SUB NeEqE
      SHARED Operation!(),NoVar%,Operation$()
      Operation$(10,NoVar%) = ""
      FOR Init% = 1 TO 9
        Operation!(Init%,NoVar%) = 1000000
      NEXT Init%
END SUB
'------------------------------------------------------------------------
SUB ClEdSc
      SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      FOR Cl% = 2 TO 4
        LOCATE Cl%,10,0
        PRINT SPACE$(66)
      NEXT Cl%
END SUB
'------------------------------------------------------------------------
Graph:
'determine the type of graphics card to use
      ON ERROR GOTO ErrorTrap
9775  SCREEN 2  'Hercules or CGA in black & white
      DEF SEG = 0
      VideoMode% = PEEK(&H449)
      VideoGraph% = VideoMode%
9780  SCREEN 8  'CGA with color
      VideoGraph% = 14
9785  SCREEN 9  'EGA
      VideoGraph% = 16
9790  ON ERROR GOTO 0
      RETURN
'------------------------------------------------------------------------
'routine used to enter an integer value
SUB EditInteger(EdValue%,EdRow%,EdCol%,EdNoChr%)
30000 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue% = CINT(VAL(EdStr$))
END SUB
'------------------------------------------------------------------------
'routine used to enter a long integer value
SUB EditLong(EdValue&,EdRow%,EdCol%,EdNoChr%)
30050 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue& = CLNG(VAL(EdStr$))
```

```
END SUB
'------------------------------------------------------------------------
'routine used to enter a real value
SUB EditReal(EdValue!,EdRow%,EdCol%,EdNoChr%)
30100 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue! = CSNG(VAL(EdStr$))
END SUB
'------------------------------------------------------------------------
'routine used to enter a string
SUB EditDouble(EdValue#,EdRow%,EdCol%,EdNoChr%)
30150 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue# = CDBL(VAL(EdStr$))
END SUB
'------------------------------------------------------------------------
'routine used to enter data
SUB EditString(EdStr$,EdRow%,EdCol%,EdNoChr%)
30200 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
END SUB
'------------------------------------------------------------------------
'this routine enters a string of up to MaxNoChr% characters
SUB EditData(EdRow%,EdCol%,EdInStr$,EdNoChr%)
30250 LOCAL LetPos%,NoChr%,Lines%,StartCol%
      EdInStr$ = STRING$(EdNoChr%,0) : LetPos% = 1
      StartCol% = EdCol%
      NoChr% = 1  :  EdCol% = StartCol%
      DO WHILE NoChr% <= EdNoChr%
        DO
          LOCATE EdRow%,EdCol%,1
        LOOP UNTIL INSTAT
        Letter$ = INKEY$
        SELECT CASE Letter$
          CASE CHR$(27)
            EXIT SUB
          CASE CHR$(13)
            EXIT LOOP
          CASE CHR$(8)
            IF EdCol% > StartCol% THEN   'avoid backspacing
              DECR EdCol% : DECR NoChr%  :  DECR LetPos%
              LOCATE EdRow%,EdCol%,0  :  PRINT SPACE$(1)
              MID$(EdInStr$,LetPos%,1) = CHR$(0)
            END IF
          CASE ELSE
            IF LEN(Letter$) = 1 THEN
              LOCATE EdRow%,EdCol%,0
              PRINT Letter$
              MID$(EdInStr$,LetPos%,1) = Letter$
              INCR EdCol% : INCR NoChr%  :  INCR LetPos%
            END IF
        END SELECT
      LOOP
      EdInStr$ = LEFT$(EdInStr$,LetPos%-1)
END SUB
```

```
'-----------------------------------------------------------------------------
' create the directions for the editing screen
SUB EditScreen
     SHARED ColArr%()
     COLOR ColArr%(12),ColArr%(15)
     LOCATE 2,10,0 : PRINT CHR$(218) + STRING$(64,196) + CHR$(191)
     LOCATE 3,10,0 : PRINT CHR$(179) + SPACE$(64) + CHR$(179)
     LOCATE 4,10,0 : PRINT CHR$(192) + STRING$(64,196) + CHR$(217)
     LOCATE 3,11,0 : PRINT " " + CHR$(25) + " " + CHR$(24) + " : vertical "
     LOCATE 3,28,0 : PRINT " " + CHR$(26) + " " + CHR$(27) + " : lateral "
     LOCATE 3,44,0
     COLOR ColArr%(20),ColArr%(10) : PRINT " ENTER ";
     COLOR ColArr%(12),ColArr%(15) : PRINT ": edit "
     COLOR ColArr%(10),ColArr%(1)
     LOCATE 3,60,0 : PRINT " ESC : exit "
     COLOR ColArr%(12),ColArr%(15)
END SUB
'-----------------------------------------------------------------------------
'this routine displays a procedure-sensitive help screen
SUB PBAhelp(OnHelp$)
      SHARED HelpFile$
13000 SELECT CASE OnHelp$
        CASE "FIELD"
          HelpFile$ = "CHAP4.DOC"
          Title$ = "PBA Index : Field Performance"
        CASE "MAIN"
          HelpFile$ = "CHAP2.DOC"
          Title$ = "Main Menu"
        CASE "SHARE"
          HelpFile$ = "CHAP7.DOC"
          Title$ = "Deterioration Proportions"
        CASE "USER"
          HelpFile$ = "CHAP5.DOC"
          Title$ = "User Equation"
        CASE "ZERO"
          HelpFile$ = "CHAP6.DOC"
          Title$ = "Level of Maintenance"
        CASE "PAVEMENT"
          HelpFile$ = "CHAP7.DOC"
          Title$ = "Pavement Sections"
        CASE "XVAR"
          HelpFile$ = "CHAP5.DOC"
          Title$ = "Independent Variable"
        CASE "RELATION"
          HelpFile$ = "CHAP5.DOC"
          Title$ = "Equations Relationship"
      END SELECT
      CALL HelpScr(HelpFile$,Title$)
END SUB
'-----------------------------------------------------------------------------
'this routine opens an ASCII file and displays its content
SUB HelpScr(HelpFile$,Title$)
```

```
      shared wrow%(),wrows%(),wcol%(),wcols%(),wattr%(),wbrdr%(),wshdw%()
      shared scrn%(),wptr(),Help$,InHelp$,FileErr$,ColArr%()
      LOCAL ViewFile$()
      InHelp$ = "Y" : FileErr$ = HelpFile$
4000  LOCATE 19,26,0
4050  OPEN HelpFile$ FOR INPUT AS # 5
      IF Help$ <> "H" THEN
        ColScr% = FNATTR%(ColArr%(5),ColArr%(6))
        CALL MsgWindow(3,5,20,70,ColScr%,2,1,0)
        Title$ = " " + Title$ + " "
        Guide$ = " ENTER : next     ESC : exit "
        CALL TITLEWINDOW(5,Guide$)
        CALL TITLEWINDOW(2,Title$)
        GoBack$ = CHR$(13)  'control variable
4200    DO   'do until the user press ESC to exit, or the file is read totally
          Row% = 0
          CALL ClearWindow
          DIM DYNAMIC ViewFile$(1:17)
          DO
            DO
              SELECT CASE GoBack$
                CASE CHR$(13)
                  INCR Row%
4250              LINE INPUT # 5, GetLine$
                  ViewFile$(Row%) = LEFT$(GetLine$,65)
                  CALL PrtWindow(Row%,2,ViewFile$(Row%))
              END SELECT
              IF Row% >= 17 OR EOF(5) THEN EXIT LOOP
            LOOP
            DO : LOOP UNTIL INSTAT
            GoBack$ = INKEY$
            IF GoBack$ = CHR$(27) THEN EXIT LOOP
          LOOP UNTIL GoBack$ = CHR$(13)
          ERASE ViewFile$
          IF GoBack$ = CHR$(27) THEN EXIT LOOP
        LOOP UNTIL EOF(5)
4275    CLOSE # 5
        CALL REMOVEWINDOW
        erase wrow%,wrows%,wcol%,wcols%,wattr%,wbrdr%,wshdw%,scrn%,wptr
      END IF
4300  Help$ = "Y"
END SUB
'---------------------------------------------------------------------------
'display the error message screen
SUB MsgWindow(ROW%,COL%,ROWS%,COLS%,ATTR%,BRDRSEL%,SHADOW%,ZOOM%)
12950 SHARED wrow%(),wrows%(),wcol%(),wcols%(),wattr%(),InHelp$
      SHARED wbrdr%(),wshdw%(),scrn%(),wptr(),ColArr%(),ErMsg$()
      mw% = 3  :  ScrnArray = 8000
      dim dynamic wrow%(mw%),wrows%(mw%),wcol%(mw%),wcols%(mw%),wattr%(mw%)
      dim dynamic wbrdr%(mw%),wshdw%(mw%),scrn%(ScrnArray),wptr(mw%)
      CALL MAKEWINDOW(ROW%,COL%,ROWS%,COLS%,ATTR%,BRDRSEL%,SHADOW%,ZOOM%)
      IF InHelp$ <> "Y" THEN
```

```
        CALL PRTCWINDOW(1,ErMsg$(3))
        CALL PRTCWINDOW(2,ErMsg$(1))
        CALL PRTCWINDOW(3,ErMsg$(2))
        CALL PRTCWINDOW(4,ErMsg$(4))
        DO
          ErMsg$(5) = INKEY$
        LOOP UNTIL ErMsg$(5) = CHR$(27)
        CALL REMOVEWINDOW
        erase wrow%,wrows%,wcol%,wcols%,wattr%,wbrdr%,wshdw%,scrn%,wptr
      END IF
      InHelp$ = ""
END SUB
'------------------------------------------------------------------------
'These utilities are released into the public domain for anyone to use as they
'see fit; however, anyone who feels inspired and would like to contribute to
'my printer paper and computer supplies fund may feel free to do so.
'Rick Fothergill
'141 Oak Hill Road
'Pittsfield, MA 01201
'(413) 442-2456
'GEnie Address: R.FOTHERGILL
'Compuserve: 76210,443
'........................................................................
sub MakeWindow(Row%,Col%,Rows%,Cols%,Attr%,BrdrSel%,Shadow%,Zoom%) static
  shared wrow%(),wrows%(),wcol%(),wcols%(),wattr%(),wbrdr%(),wshdw%(),scrn%()
  shared wptr(),LI,mw%,ScrnArray
  local r1%,r2%,c1%,c2%,colratio%,wsize
  select case shadow%
    Rem Left
    case = 1
      c1%=col%-2 : c2%=cols%+2 : r2%=rows%+1
    Rem Right
    case = 2
      c1%=col%   : c2%=cols%+2 : r2%=rows%+1
    case else
      c1%=col%   : c2%=cols%   : r2%=rows%
  end select
  wsize = (r2% * c2%) * 2
  LI = LI + 1
  Wptr(LI+1) = Wptr(LI)+WSize+1
  WRow%(LI)  = Row%
  WCol%(LI)  = Col%
  WRows%(LI) = Rows%
  WCols%(LI) = Cols%
  Wattr%(LI) = Attr%
  WBrdr%(LI) = BrdrSel%
  WShdw%(LI) = Shadow%
  Call Qsave(Row%,c1%,r2%,c2%,scrn%(Wptr(LI)))
  if zoom% = 1 then
    r1% = row% + (rows%\2)
    r2% = row% + rows%-(rows%\2)
    c1% = col% + (cols%\2)
```

```
    c2% = col% + cols%-(cols%\2)
    colratio% = (cols% \ rows%)+1
    if colratio% > 4 then colratio%=4
    do
      if r1%>row% then r1%=r1%-1
      if r2%<(row%+rows%) then r2%=r2%+1
      if c1%>col% then c1%=c1%-colratio%
      if c1%<col% then c1%=col%
      if c2%<(col%+cols%) then c2%=c2%+colratio%
      if c2%>(col%+cols%) then c2%=col%+cols%
      call Qbox(r1%,c1%,r2%-r1%,c2%-c1%,attr%,brdrsel%)
    loop until c1%=col% and c2%=col%+cols% and r1%=row% and r2%=row%+rows%
  else
    call Qbox(row%,col%,rows%,cols%,attr%,brdrsel%)
  end if
  select case shadow%
    rem Left
    case = 1
      call qfill(row%+1,col%-2,rows%-1,2,asc(" "),0)
      call qfill(row%+rows%,col%-2,1,cols%,asc(" "),0)
    rem Right
    case = 2
      call qfill(row%+1,col%+cols%,rows%-1,2,asc(" "),0)
      call qfill(row%+rows%,col%+2,1,cols%,asc(" "),0)
    case else
  end select
end sub

sub TitleWindow(dir%,title$) static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
  select case dir%
    rem UpperLeft
    case = 1
      call qprint(wrow%(LI),wcol%(LI)+2,title$,wattr%(LI))
    rem UpperCenter
    case = 2
      call qprintc(wrow%(LI),wcol%(LI),wcol%(LI)+wcols%(LI)-1,_
                   title$,wattr%(LI))
    rem UpperRight
    case = 3
      call qprint(wrow%(LI),wcol%(LI)+wcols%(LI)-len(title$)-2,_
                  title$,wattr%(LI))
    rem LowerLeft
    case = 4
      call qprint(wrow%(LI)+wrows%(LI)-1,wcol%(LI)+2,title$,wattr%(LI))
    rem LowerCenter
    case = 5
      call qprintc(wrow%(LI)+wrows%(LI)-1,wcol%(LI),wcol%(LI)+wcols%(LI)-1,_
                   title$,wattr%(LI))
    rem LowerRight
    case = 6
      call qprint(wrow%(LI)+wrows%(LI)-1,wcol%(LI)+wcols%(LI)-len(title$)-2,_
```

```
                    title$,wattr%(LI))
    case else
  end select
end sub

sub RemoveWindow static
shared Wrow%(),WCol%(),WRows%(),Wcols%(),Wattr%(),WShdw%(),Scrn%(),Wptr(),LI
  if LI = 0 then
    print "NO WINDOW TO REMOVE"
  else
    select case WShdw%(LI)
    case = 1
      call qrest(Wrow%(LI),WCol%(LI)-2,WRows%(LI)+1,WCols%(LI)+2,_
                 Scrn%(Wptr(LI)))
    case = 2
      call qrest(WRow%(LI),WCol%(LI),WRows%(LI)+1,WCols%(LI)+2,Scrn%(Wptr(LI)))
    case else
      call qrest(WRow%(LI),Wcol%(LI),WRows%(LI),WCols%(LI),Scrn%(Wptr(LI)))
    end select
    LI = LI -1
  end if
end sub

sub Qbox(Row%,Col%,Rows%,Cols%,attr%,BrdrSel%) static
  if rows%>2 and cols%>2 then
    if brdrsel% > 0 and brdrsel% < 6 then
      on brdrsel% gosub single,double,mixed12,mixed21,doubleleftarrow
      call qprint(row%,col%,tl$,attr%)
      call qfill (row%,col%+1,1,cols%-2,asc(th$),attr%)
      call qprint(row%,col%+cols%-1,tr$,attr%)
      call qfill (row%+1,col%,rows%-2,1,asc(lv$),attr%)
      call qfill (row%+1,col%+cols%-1,rows%-2,1,asc(rv$),attr%)
      call qprint(row%+rows%-1,Col%,bl$,attr%)
      call qfill (row%+rows%-1,Col%+1,1,cols%-2,asc(bh$),attr%)
      call qprint(row%+rows%-1,col%+cols%-1,br$,attr%)
      call qfill (row%+1,col%+1,rows%-2 ,cols%-2,asc(" "),attr%)
    else
      call qfill (row%,col%,rows%,cols%,asc(" "),attr%)
    end if
  end if
  exit sub

Single:
  TL$=CHR$(218):TH$=CHR$(196):TR$=CHR$(191)
  LV$=CHR$(179):RV$=CHR$(179)
  BL$=CHR$(192):BH$=CHR$(196):BR$=CHR$(217)
  Return
Double:
  TL$=CHR$(201):TH$=CHR$(205):TR$=CHR$(187)
  LV$=CHR$(186):RV$=CHR$(186)
  BL$=CHR$(200):BH$=CHR$(205):BR$=CHR$(188)
  Return
```

```
Mixed12:
  TL$=CHR$(214):TH$=CHR$(196):TR$=CHR$(183)
  LV$=CHR$(186):RV$=CHR$(186)
  BL$=CHR$(211):BH$=CHR$(196):BR$=CHR$(189)
  Return
Mixed21:
  TL$=CHR$(213):TH$=CHR$(205):TR$=CHR$(184)
  LV$=CHR$(179):RV$=CHR$(179)
  BL$=CHR$(212):BH$=CHR$(205):BR$=CHR$(190)
  Return
DoubleLeftArrow:
  TL$=CHR$(17):TH$=CHR$(205):TR$=CHR$(187)
  LV$=CHR$(186):RV$=CHR$(186)
  BL$=CHR$(200):BH$=CHR$(205):BR$=CHR$(188)
  Return
end sub

sub ClearWindow static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
   call qfill(wrow%(LI)+1,wcol%(LI)+1,wrows%(LI)-2,wcols%(LI)-2,asc(" "),_
             wattr%(LI))
end sub

sub PrtWindow(row%,col%,StrDat$) static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
   call qprint(wrow%(LI)+row%,wcol%(LI)+col%,StrDat$,wattr%(LI))
end sub

sub PrtCWindow(row%,StrDat$) static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
   call qprintc(wrow%(LI)+row%,wcol%(LI),wcol%(LI)+wcols%(LI),StrDat$,_
               wattr%(LI))
end sub

def fnattr%(fore%,back%)
  local temp%
  temp%=(back%*16)+fore%
  if fore%>15 then temp% = temp% + 112
  fnattr% = temp%
end def

SUB QPRINT INLINE
  $INLINE "QPRINT.BIN"
END SUB
rem CALL QPRINT(ROW%,COL%,STR$,ATTR%)

SUB QPRINTC INLINE
  $INLINE "QPRINTC.BIN"
END SUB
rem CALL QPRINTC(ROW%,COLL%,COLR%,STRDAT$,ATTR%)

SUB QFILL INLINE
```

```
  $INLINE "QFILL.BIN"
END SUB
rem CALL QFILL(ROW%,COL%,ROWS%,COLS%,CHAR%,ATTR%)

SUB QATTR INLINE
  $INLINE "QATTR.BIN"
END SUB
rem CALL QATTR(ROW%,COL%,ROWS%,COLS%,ATTR%)

SUB QSAVE INLINE
  $INLINE "QSAVE.BIN"
END SUB
rem CALL QSAVE(ROW%,COL%,ROWS%,COLS%,SCRN%(??))

SUB QREST INLINE
  $INLINE "QREST.BIN"
END SUB
rem CALL QREST(ROW%,COL%,ROWS%,COLS%,SCRN%(??))
'---------------------------------------------------------------------------
```

Appendix A.5
PBAEQN.BAS

```
'-----------------------------------------------------------------------------
'
'                              PBAEQN.BAS
'
'the following program was coded using Turbo BASIC version 1.1.  This file
'contains the subroutines used during the execution of the main file
'PBAASHTO.BAS, or PBAFIELD.BAS, to enter the performance equation defined by
'the user.
'-----------------------------------------------------------------------------
'break source code in segments of 64K
      $SEGMENT
'turn key trapping off
      $EVENT OFF
'-----------------------------------------------------------------------------
'a file to store the data which form the performance equations is to be
'created next, a random access file is to be used as the data base
SUB UserEqn(OnField$)
      SHARED Help$,NewEq$,Viewing$,OnHelp$,Exists$,AreaBetween!,PBAindex$,EqID$
      SHARED XvUnit$,UnitsX$,PSI!(),PerformanceM$,PfUnit$,PfDesign!,Ymax!,Ymin!
      SHARED HighwayNo$,EqAddr%(),FileEqn$,ColArr%(),Xval!(),Yval!(),EqPrint$
      SHARED NewName$
      DO
17000   CALL FileScreen
        NewName$ = "N" : Help$ = "" : NewEq$ = "" : OnHelp$ = "USER"  'flags
        'create the menu for file creation
17050   LOCATE 4,21,0  :  PRINT " Performance Equation - Data File "
        LOCATE 6,15,0  :  PRINT "1. Create a new Equation File"
        LOCATE 7,15,0  :  PRINT "2. Use an existent Equation File"
        LOCATE 8,15,0  :  PRINT "3. View an Equation File"
        LOCATE 9,15,0  :  PRINT "4. Return to Previous Menu"
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 10,15,0  :  PRINT " Enter a choice : ";
        DO
          Valid$ = "N"
          CALL MenuChoice(Choice$,10,34)
          SELECT CASE OnField$
            CASE "N","F"
              SELECT CASE Choice$
                CASE CHR$(49),CHR$(50),CHR$(51),CHR$(52)
                  Valid$ = "Y"
                  Viewing$ = "N"
              END SELECT
            CASE "Y"
              SELECT CASE Choice$
                CASE CHR$(50),CHR$(51),CHR$(52)
                  Valid$ = "Y"
                  Viewing$ = "Y"
              END SELECT
          END SELECT
        LOOP UNTIL Valid$ = "Y"
        FileChoice% = VAL(Choice$)
        SELECT CASE FileChoice%
```

```
        CASE 1   ' to create a brand new performance equation file
          LOCATE 4,21,0  :  PRINT "  New Performance Equation File  "
          EquationSpec$ = " created"
        CASE 2   ' to use an existent performance equation file
          LOCATE 4,21,0  :  PRINT " Existent Performance Equation File "
          EquationSpec$ = " opened"
        CASE 3
          LOCATE 4,21,0  :  PRINT "  View Performance Equation File  "
          EquationSpec$ = " seen"  :  Viewing$ = "Y"
        CASE 4
          Help$ = "C"
      END SELECT
      IF Help$ <> "C" THEN
17105   COLOR ColArr%(15),ColArr%(1)
        LOCATE 6,13,0
        PRINT "Enter name of performance equation file to be" + EquationSpec$
        'check the existence of the file
        FileExt$ = ".EQN"
        IF INSTR(1,"FY",OnField$) <> 0 THEN FileExt$ = ".FEQ"
        CALL FiNaCheck(FileExt$,FileChoice%)
        IF NewName$ <> "Y" THEN CLOSE # 4
        IF NewName$ = "Y" THEN
          CALL CleanScr
        ELSE
          IF FileChoice% = 3 THEN
            CALL OpenFile(FileEqn$)
            CALL ViewEqn(FileEqn$)
            CLOSE # 4
            NewName$ = "Y"
          ELSE
            'if from the Predicted Serviceability Module
            IF INSTR(1,"NF",OnField$) <> 0 THEN
              'create the menu for equation creation
              DO
                Exists$ = "Y"
17110           CALL CleanScr
17115           LOCATE 4,21,0
                PRINT "        Performance Equation           "
                LOCATE 6,15,0
                PRINT "1. Create a new Performance Equation" + SPACE$(13)
                LOCATE 7,15,0
                PRINT "2. Use an existent Performance Equation" + SPACE$(11)
                LOCATE 8,15,0
                PRINT "3. Return to previous menu" + SPACE$(15)
                LOCATE 9,15,0  :  PRINT SPACE$(50)
                LOCATE 10,15,0 :  PRINT SPACE$(45)
                LOCATE 10,15,0 :  PRINT " Enter a choice : ";
                DO
                  CALL MenuChoice(Choice$,10,34)
                  SELECT CASE Choice$
                    CASE CHR$(49),CHR$(50),CHR$(51)
                      EXIT LOOP
```

```
            END SELECT
          LOOP
          EquationChoice% = VAL(Choice$)
          'save the address of number of equations
          DIM DYNAMIC EqAddr%(1:4)
17130     SELECT CASE EquationChoice%
            CASE 1   ' to create a brand new performance equation
              CALL OpenFile(FileEqn$)
              CALL SubEquation(EqAddr%())
              IF Help$ <> "Q" THEN
                CALL NewEquation(EqAddr%())
                IF Help$ <> "Q" THEN
                  IF OnField$ = "F" THEN
                    'enter the data for the X axis
                    CALL InputData("T")
                  END IF
                  IF Help$ <> "Q" THEN
                    CALL PefWait("Y")
                    CALL PfIndex(AreaBetween!)
                    'save the last subequation in the data base file
                    LSET PBAindex$ = MKS$(AreaBetween!)
                    LSET XvUnit$ = UnitsX$
                    EqSet& = INT((LOF(4)/472) + 1)
                    PUT # 4, EqSet&
                    IF OnField$ = "N" THEN Help$ = "S"
                  ELSE
                    ERASE PSI!
                  END IF
                END IF
              END IF
            CASE 2   ' to use an existent performance equation
              CALL OpenFile(FileEqn$)
              CALL SubEquation(EqAddr%())
              IF Help$ <> "Q" THEN
                CALL OldEquation(EqAddr%())
                IF Help$ <> "Q" THEN
                  IF OnField$ = "F" THEN
                    'enter the data for the X axis
                    CALL InputData("T")
                  END IF
                  IF Help$ <> "Q" THEN
                    CALL PefWait("Y")
                    CALL PfIndex(AreaBetween!)
                    'save the last subequation in the data base file
                    LSET PBAindex$ = MKS$(AreaBetween!)
                    LSET XvUnit$ = UnitsX$
                    EqSet& = INT((LOF(4)/472) + 1)
                    PUT # 4, EqSet&
                    IF OnField$ = "N" THEN Help$ = "S"
                  ELSE
                    ERASE PSI!
                  END IF
```

```
                        ELSE
                          CLOSE # 4
                        END IF
                      END IF
                    CASE 3
                      Help$ = "Q"
                      Exists$ = "F"
                  END SELECT
                  ERASE EqAddr%
                  IF Help$ <> "Q" THEN
                    CLOSE # 4
                    IF OnField$ = "F" THEN
                      'display PBA index and plot graphic
                      CALL PBAindex
                    END IF
                    Exists$ = "N"
                  END IF
                  IF Help$ = "S" THEN EXIT LOOP
                  IF Exists$ = "N" THEN CALL FileScreen
                LOOP UNTIL Exists$ <> "N"
17135           IF Help$ = "Q" THEN
                  NewName$ = "Y"
                  CLOSE # 4
                END IF
              ELSEIF OnField$ = "Y" THEN
                'if from the Field Performance Module
                Help$ = "S"
              END IF
            END IF
          END IF
        END IF
        IF INSTR(1,"CS",Help$) <> 0 THEN EXIT LOOP
      LOOP UNTIL NewName$ <> "Y"
END SUB
'-------------------------------------------------------------------------------
'the following routine is used to enter the identification data and the
'number of subequations for each main equation
SUB SubEquation(EqAddr%(1))
      SHARED PerformanceM$,PfUnit$,PfDesign!,Ymax!,Ymin!,HighwayNo$,EqID$
      SHARED ColArr%(),Help$,EqPrint$,Editing$,EditEq$
      Editing$ = "Eq"
      COLOR ColArr%(15),ColArr%(1)
      CLS
      LOCATE 3,5,0   :  PRINT CHR$(201) + STRING$(68,205) + CHR$(187)
      FOR Frame% = 4 TO 17 STEP 1
        LOCATE Frame%,5,0  :  PRINT CHR$(186) + STRING$(68,0) + CHR$(186)
      NEXT Frame%
      LOCATE 18,5,0    :  PRINT CHR$(204) + STRING$(68,205) + CHR$(185)
      LOCATE 19,5,0    :  PRINT CHR$(186) + STRING$(68,0) + CHR$(186)
      LOCATE 20,5,0    :  PRINT CHR$(200) + STRING$(68,205) + CHR$(188)
      LOCATE 7,15,0    :  PRINT "Enter information for MAIN EQUATION :"
      LOCATE 9,20,0    :  PRINT "Performance Measure       :"
```

```
LOCATE 10,20,0  :  PRINT "Performance Units           :"
LOCATE 11,20,0  :  PRINT "Performance-Design Value    :"
LOCATE 12,20,0  :  PRINT "Maximum Performance Scale :"
LOCATE 13,20,0  :  PRINT "Minimum Performance Scale :"
LOCATE 14,20,0  :  PRINT "Pavement Section            :"
LOCATE 15,20,0  :  PRINT "Equation ID Number          :"
LOCATE 16,20,0  :  PRINT "Number of Sub-Equations   "
LOCATE 17,20,0  :  PRINT "               (2 minimum)    :"
COLOR ColArr%(12),ColArr%(15)
LOCATE 5,22,0  :  PRINT " User-Defined Performance Equation "
LOCATE 19,27,0 :  PRINT " F3 : edit "
LOCATE 19,39,0 :  PRINT " ESC : quit "
DO
  EditEq$ = "N"  :  Help$ = "N"
  COLOR ColArr%(12),ColArr%(15)
  FOR Box% = 9 TO 15
    LOCATE Box%,48,0  :  PRINT SPACE$(15)
  NEXT Box%
  LOCATE 17,48,0  :  PRINT SPACE$(15)
  'enter the information required
  CALL InputString(StringIN$,9,49,13)
  IF Help$ <> "Q" THEN
    PerformanceM$ = StringIn$
    CALL InputString$(StringIN$,10,49,13)
    IF Help$ <> "Q" THEN
      PfUnit$ = StringIN$
      CALL InputReal(Value!,11,49,10)
      IF Help$ <> "Q" THEN
        PfDesign! = Value!
        CALL InputReal(Value!,12,49,10)
        IF Help$ <> "Q" THEN
          Ymax! = Value!
          CALL InputReal(Value!,13,49,10)
          IF Help$ <> "Q" THEN
            Ymin! = Value!
            CALL InputString(StringIn$,14,49,13)
            IF Help$ <> "Q" THEN
              HighwayNo$ = StringIn$
              CALL InputString(StringIn$,15,49,10)
              IF Help$ <> "Q" THEN
                EqID$ = StringIn$
                EqPrint$ = EqID$  'to print the ID on the screen
                DO
                  DO
                    LOCATE 17,49,0  :  PRINT SPACE$(2)
                    CALL InputInteger(Value%,17,49,2)
                    IF Help$ = "Q" THEN EXIT LOOP
                  LOOP UNTIL Value% >= 2
                  IF Help$ = "Q" THEN EXIT LOOP
                LOOP UNTIL Value% <= 5
                IF Help$ <> "Q" THEN
                  EqAddr%(1) = Value%
```

```
                    Editing$ = ""
                  END IF
                END IF
              END IF
            END IF
          END IF
        END IF
      END IF
    END IF
  LOOP UNTIL EditEq$ = "N"
END SUB
'------------------------------------------------------------------------------
'the following subroutine is used to retrieve a performance equation from
'the data base
SUB OldEquation(EqAddr%(1))
  SHARED OldEq$,EqPrint$,Exists$,ColArr%(),Help$,DescrVar$,NoVar$,NoVal%
  SHARED SuEqVa!(),OnHelp$,Operation!(),Values!(),NoVar%,Neck$,Operation$()
  SHARED NoData%,PfDesign!,SubEqId$,PerformanceM$,HighwayNo$,Dat!(),PSI!()
  SHARED VarDesc1$,VarCoef1$,LogVar1$,Var101$,LnVar1$,eVar1$,ConsVar1$
  SHARED RootVar1$,ExpVar1$,VarConst1$,VarOp1$,VarDesc2$,VarCoef2$,LogVar2$
  SHARED Var102$,LnVar2$,eVar2$,ConsVar2$,RootVar2$,ExpVar2$,VarConst2$
  SHARED VarOp2$,VarDesc3$,VarCoef3$,LogVar3$,Var103$,LnVar3$,eVar3$,XvUnit$
  SHARED ConsVar3$,RootVar3$,ExpVar3$,VarConst3$,VarOp3$,VarDesc4$,VarCoef4$
  SHARED LogVar4$,Var104$,LnVar4$,eVar4$,ConsVar4$,RootVar4$,ExpVar4$
  SHARED VarConst4$,VarOp4$,VarDesc5$,VarCoef5$,LogVar5$,Var105$,LnVar5$
  SHARED eVar5$,ConsVar5$,RootVar5$,ExpVar5$,VarConst5$,VarOp5$,PfUnit$
  SHARED Performance$,Highway$,EquationID$,PerfDesign$,PBAindex$,FileEqn$
  'check that all of the subequations exist
  EndFile& = LOF(4)/472
  EqAddr%(3) = 1
  DO
    OldEq$ = "Y" : SubEqId$ = "" : Help$ = "N"
    Lth% = LEN(EqPrint$)
    EqAddr%(4) = 48 + EqAddr%(3)
    SubEqId$ = LEFT$(EqPrint$,Lth%) + CHR$(45) + CHR$(EqAddr%(4))
    Lth% = LEN(SubEqId$)
    FOR Equation& = 1 TO EndFile&
      Exists$ = "Y"
      GET # 4, Equation&
      IF LEFT$(SubEqId$,Lth%) = LEFT$(EquationID$,Lth%) THEN EXIT FOR
      Exists$ = "N"
    NEXT Equation&
    INCR EqAddr%(3)
    IF Exists$ = "N" THEN EXIT LOOP
  LOOP UNTIL EqAddr%(3) > EqAddr%(1)
  IF Exists$ = "N" THEN
    COLOR ColArr%(15),ColArr%(1)
    LOCATE 17,5,0  :  PRINT CHR$(201) + STRING$(68,205) + CHR$(187)
    FOR Frame% = 18 TO 20
      LOCATE Frame%,5,0  :  PRINT CHR$(186) + STRING$(68,0) + CHR$(186)
    NEXT Frame%
    LOCATE 21,5,0  :  PRINT CHR$(200) + STRING$(68,205) + CHR$(188)
```

```
    LOCATE 18,20,0  :  PRINT "The Equation ";
    COLOR ColArr%(12),ColArr%(15)
    PRINT USING " \          \";SubEqId$;
    COLOR ColArr%(15),ColArr%(1)
    PRINT " does not exist "
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 20,25,0  :  PRINT " Press any key to continue "
    DO  :  LOOP UNTIL INSTAT
    Quit$ = INKEY$
    Help$ = "Q"
  END IF
  'call the equation if it exists
  IF Exists$ = "Y" THEN
    'save the results of the performance subroutine
    DIM DYNAMIC SuEqVa!(1:10,1:5)
    EqAddr%(2) = 1 : EqAddr%(4) = 0
    CLOSE # 4
    CALL OpenFile(FileEqn$)
    DO
      OnHelp$ = "USER"
      'get the equation
      EqAddr%(4) = 48 + EqAddr%(2)
      SubEqId$ = LEFT$(EqPrint$,Lth%) + CHR$(45) + CHR$(EqAddr%(4))
      Lth% = LEN(SubEqId$)
      FOR Equation& = 1 TO EndFile&
        GET # 4, Equation&
        IF LEFT$(SubEqId$,Lth%) = LEFT$(EquationID$,Lth%) THEN EXIT FOR
      NEXT Equation&
      DIM DYNAMIC Operation!(1:10,1:5),Values!(1:10,1:5)
      DIM DYNAMIC Operation$(10:10,1:5)
      FOR Init1% = 1 TO 10  'initialize the matrix to ones
        FOR Init2% = 1 TO 5
          Values!(Init1%,Init2%) = 1
        NEXT Init2%
      NEXT Init1%

      LSET Performance$ = PerformanceM$
      LSET PerfDesign$ = MKS$(PfDesign!)
      LSET Highway$ = HighwayNo$
      LSET EquationID$ = EquationID$
      LSET NoVar$ = NoVar$
      LSET PefUnits$ = PfUnit$
      LSET XvUnit$ = XvUnit$
      'store the operators in the Operation! array
      DescrVar$ = VarDesc1$
      Operation!(1,1) = CVS(VarCoef1$)  : Operation!(2,1) = CVS(LogVar1$)
      Operation!(3,1) = CVS(Var101$)    : Operation!(4,1) = CVS(LnVar1$)
      Operation!(5,1) = CVS(eVar1$)     : Operation!(6,1) = CVS(ConsVar1$)
      Operation!(7,1) = CVS(RootVar1$)  : Operation!(8,1) = CVS(ExpVar1$)
      Operation!(9,1) = CVS(VarConst1$) : Operation$(10,1) = VarOp1$
      CALL SaveData(1)
```

```
DescrVar$ = VarDesc2$
Operation!(1,2) = CVS(VarCoef2$)   : Operation!(2,2) = CVS(LogVar2$)
Operation!(3,2) = CVS(Var102$)     : Operation!(4,2) = CVS(LnVar2$)
Operation!(5,2) = CVS(eVar2$)      : Operation!(6,2) = CVS(ConsVar2$)
Operation!(7,2) = CVS(RootVar2$)   : Operation!(8,2) = CVS(ExpVar2$)
Operation!(9,2) = CVS(VarConst2$)  : Operation$(10,2) = VarOp2$
CALL SaveData(2)

DescrVar$ = VarDesc3$
Operation!(1,3) = CVS(VarCoef3$)   : Operation!(2,3) = CVS(LogVar3$)
Operation!(3,3) = CVS(Var103$)     : Operation!(4,3) = CVS(LnVar3$)
Operation!(5,3) = CVS(eVar3$)      : Operation!(6,3) = CVS(ConsVar3$)
Operation!(7,3) = CVS(RootVar3$)   : Operation!(8,3) = CVS(ExpVar3$)
Operation!(9,3) = CVS(VarConst3$)  : Operation$(10,3) = VarOp3$
CALL SaveData(3)

DescrVar$ = VarDesc4$
Operation!(1,4) = CVS(VarCoef4$)   : Operation!(2,4) = CVS(LogVar4$)
Operation!(3,4) = CVS(Var104$)     : Operation!(4,4) = CVS(LnVar4$)
Operation!(5,4) = CVS(eVar4$)      : Operation!(6,4) = CVS(ConsVar4$)
Operation!(7,4) = CVS(RootVar4$)   : Operation!(8,4) = CVS(ExpVar4$)
Operation!(9,4) = CVS(VarConst4$)  : Operation$(10,4) = VarOp4$
CALL SaveData(4)

DescrVar$ = VarDesc5$
Operation!(1,5) = CVS(VarCoef5$)   : Operation!(2,5) = CVS(LogVar5$)
Operation!(3,5) = CVS(Var105$)     : Operation!(4,5) = CVS(LnVar5$)
Operation!(5,5) = CVS(eVar5$)      : Operation!(6,5) = CVS(ConsVar5$)
Operation!(7,5) = CVS(RootVar5$)   : Operation!(8,5) = CVS(ExpVar5$)
Operation!(9,5) = CVS(VarConst5$)  : Operation$(10,5) = VarOp5$
CALL SaveData(5)

FOR NoVar% = 1 TO CVI(NoVar$)
  SELECT CASE NoVar%
    CASE 1
      Neck$ = LEFT$(VarDesc1$,25)
    CASE 2
      Neck$ = LEFT$(VarDesc2$,25)
    CASE 3
      Neck$ = LEFT$(VarDesc3$,25)
    CASE 4
      Neck$ = LEFT$(VarDesc4$,25)
    CASE 5
      Neck$ = LEFT$(VarDesc5$,25)
  END SELECT
  Cons$ = LEFT$(Neck$,5)
  NoVal% = 10 : AreaBetween! = 0 : Null$ = STRING$(4,48)
  IF UCASE$(LEFT$(Neck$,8)) <> "CONSTANT"_
     AND UCASE$(LEFT$(Neck$,5)) <> "DUMMY" THEN
    'to enter the different values to be substituted for each variable
    CALL InputData("E")
    IF Help$ <> "Q" THEN
```

```
            FOR Value% = 1 TO 10
              Values!(Value%,NoVar%) = Dat!(Value%,2)
            NEXT Value%
            'use the smallest number of data
            IF NoVal% > NoData% THEN NoVal% = NoData%
            ERASE Dat!
          ELSE
            ERASE Operation!,Values!,Operation$
          END IF
        END IF
        IF Help$ = "Q" THEN EXIT FOR
17600 NEXT NoVar%
      IF Help$ <> "Q" THEN
        FOR S% = NoVar% TO 5     'don't use the remaining variables
          FOR RowVal% = 1 TO 10
            Values!(RowVal%,S%) = 0
          NEXT RowVal%
        NEXT S%
        'compute the values of performance for each subequation
        CALL Evaluate(Values!(),Operation!(),SuEqVa!(),NoVal%,EqAddr%())
        IF EqAddr%(2) < EqAddr%(1) THEN
          LSET PBAindex$ = Null$
          'save the data in the data base file
          EqSet& = INT((LOF(4)/472) + 1)
          PUT # 4, EqSet&
        END IF
        ERASE Operation!,Values!,Operation$
        INCR EqAddr%(2)
      END IF
      IF Help$ = "Q" OR Exists$ = "N" THEN EXIT LOOP
    LOOP UNTIL EqAddr%(2) > EqAddr%(1)
    IF Help$ = "Q" THEN ERASE SuEqVa!
  END IF
  IF Help$ <> "Q" AND Exists$ <> "N" THEN
    IF EqAddr%(1) > 1 THEN
      CALL EqRel
      CALL InRel
    END IF
    'compute the performance for the main equation, based on the results
    'of the subequations
    IF Help$ <> "Q" THEN
      DIM DYNAMIC PSI!(0:10)
      CALL PerMain(PSI!(),NoVal%)
      PSI!(0) = PfDesign!
      ERASE SuEqVa!
    END IF
    IF Help$ = "Q" THEN ERASE SuEqVa!
  END IF
END SUB
'------------------------------------------------------------------------
'the following equation is used to create a new performance equation
SUB NewEquation(EqAddr%(1))
```

```
SHARED SuEqVa!(),EqScr$(),OnHelp$,NoVal%,Values!(),Operation!(),ColArr%()
SHARED PerformanceM$,HighwayNo$,EqPrint$,SubEqId$,Editing$,Operation$()
SHARED NoVar%,VarID$,NewVar$,UnitsX$,Help$,DescrVar$,Neck$,Dat!(),NoData%
SHARED SecondVar$,LastVar%,EndVar$,IDvar$,Performance$,PerfDesign$,PSI!()
SHARED PfDesign!,Highway$,EquationID$,NoVar$,PefUnits$,PfUnit$,PBAindex$
SHARED XvUnit$
'save the results of the performance subroutine
 DIM DYNAMIC SuEqVa!(1:10,1:5),EqScr$(8:19)
 EqScr$(8)  = "Variable Description :"
 EqScr$(9)  = "Equation Elements    :"
 EqScr$(10) = " Coefficient         :"
 EqScr$(11) = " Log (variable)      :"
 EqScr$(12) = " 10^(variable)       :"
 EqScr$(13) = " Ln (variable)       :"
 EqScr$(14) = " e^(variable)        :"
 EqScr$(15) = " variable * constant :"
 EqScr$(16) = " Constant^(variable) :"
 EqScr$(17) = " (variable)^(exp)    :"
 EqScr$(18) = " (variable)+constant :"
 EqScr$(19) = " Variable relation   :"

 CALL EqScr
 AreaBetween! = 0 : Null$ = STRING$(4,48) : OnHelp$ = "USER"
 EqAddr%(2) = 1 : EqAddr%(4) = 0
 NoVal% = 10  'counter Evaluate routine
 DO
   'enter the information required; first, create the matrix which will
   'save the values to be substituted for each variable in the equation
   DIM DYNAMIC Values!(1:10,1:5)
   FOR Init1% = 1 TO 10                'initialize the matrix
     FOR Init2% = 1 TO 5
       Values!(Init1%,Init2%) = 1
     NEXT Init2%
   NEXT Init1%
   'create an array to save the different variables to be used during the
   'computation of the performance-ESAL index
   DIM DYNAMIC Operation!(1:10,1:5),Operation$(10:10,1:5)
   InVal& = 1000000
   FOR I% = 1 TO 10                'initialize the elements of Operation
     IF I% = 10 THEN InVal& = 0
     FOR J% = 1 TO 5
       Operation!(I%,J%) = InVal&
     NEXT J%
   NEXT I%

   COLOR ColArr%(12),ColArr%(10)
   LOCATE 3,45,0  :  PRINT USING " \                \ "; PerformanceM$
   LOCATE 4,45,0  :  PRINT USING " \                \ "; HighwayNo$
   Lth% = LEN(EqPrint$)
   EqAddr%(4) = INT(48 + EqAddr%(2))
   SubEqId$ = LEFT$(EqPrint$,Lth%)
   SubEqId$ = SubEqId$ + CHR$(45) + CHR$(EqAddr%(4))
```

```
        LOCATE 5,45,0  :  PRINT USING " \            \ "; SubEqId$
        Editing$ = "E"  'flag for EnterData routine
        FOR NoVar% = 1 TO 5
17625     DO
            COLOR ColArr%(15),ColArr%(1)
            LOCATE 20,10,0 : PRINT SPACE$(50)
            VarId$ = RIGHT$(STR$(NoVar%),1)
            COLOR ColArr%(12),ColArr%(10)
            LOCATE 6,45,0  : PRINT USING "  &  "; VarId$
            LOCATE 8,45,0  : PRINT SPACE$(25)
            LOCATE 10,45,0 : PRINT SPACE$(12)
            LOCATE 11,45,0 : PRINT SPACE$(3)
            LOCATE 12,45,0 : PRINT SPACE$(3)
            LOCATE 13,45,0 : PRINT SPACE$(3)
            LOCATE 14,45,0 : PRINT SPACE$(3)
            LOCATE 15,45,0 : PRINT SPACE$(12)
            LOCATE 16,45,0 : PRINT SPACE$(12)
            LOCATE 17,45,0 : PRINT SPACE$(12)
            LOCATE 18,45,0 : PRINT SPACE$(12)
            LOCATE 19,45,0 : PRINT SPACE$(12)
            NewVar$ = ""  'flag
            UnitsX$ = ""
17626       CALL InputString(StringIN$,8,46,24)
            IF Help$ <> "Q" AND Help$ <> "E" THEN
              DescrVar$ = StringIN$                'variable description
              CALL InputReal(Value!,10,46,10)
              IF Help$ <> "Q" AND Help$ <> "E" THEN
                IF Value! <> 0 THEN Operation!(1,NoVar%) = Value!
                IF UCASE$(DescrVar$) <> "CONSTANT" THEN
                  VarX% = 10
                  MathChosen$ = ""
                  DIM DYNAMIC MathVar$(4)
                  CALL VarDown(VarX%)
17650             DO
                    DO
                      DO : LOOP UNTIL INSTAT
                      Math$ = INKEY$
                      IF VarX% >= 15 THEN EXIT LOOP
                      SELECT CASE Math$
                        CASE CHR$(13)
17651                     LOCATE VarX%,46,0 : PRINT "x"
                          MathVar$(VarX%-11) = "x"
                          MathChosen$ = "Y"
                          EXIT LOOP
                        CASE ELSE
                          SELECT CASE MID$(Math$,2,1)
                            CASE CHR$(72),CHR$(80)
                              CALL VarDown(VarX%)
                            CASE ELSE
                              VarX% = 11
                          END SELECT
                      END SELECT
```

```
                  LOOP
                  IF (VarX%+1) > 15 OR MathChosen$ = "Y" THEN EXIT LOOP
                LOOP
17700           IF MathVar$(0) = "x" THEN
                  Operation!(2,NoVar%) = 2000000
                ELSEIF MathVar$(1) = "x" THEN
                  Operation!(3,NoVar%) = 3000000
                ELSEIF MathVar$(2) = "x" THEN
                  Operation!(4,NoVar%) = 4000000
                ELSEIF MathVar$(3) = "x" THEN
                  Operation!(5,NoVar%) = 5000000
                END IF
                ERASE MathVar$
17701           CALL InputReal(Value!,15,46,10)
                IF Help$ <> "Q" AND Help$ <> "E" THEN
                  IF Value! <> 0 THEN Operation!(6,NoVar%) = Value!
17702             CALL InputReal(Value!,16,46,10)
                  IF Help$ <> "Q" AND Help$ <> "E" THEN
                    IF Value! <> 0 THEN Operation!(7,Novar%) = Value!
17703               CALL InputReal(Value!,17,46,10)
                    IF Help$ <> "Q" AND Help$ <> "E" THEN
                      IF Value! <> 0 THEN Operation!(8,NoVar%) = Value!
17704                 CALL InputReal(Value!,18,46,10)
                      IF Help$ <> "Q" AND Help$ <> "E" THEN
                        IF Value! <> 0 THEN Operation!(9,NoVar%) = Value!
                        CALL InputString(StringIN$,19,46,10)
                        Operation$(10,NoVar%) = StringIN$
                      END IF
                    END IF
                  END IF
                END IF
              END IF
            END IF
            IF Help$ <> "Q" AND Help$ <> "E" THEN
17750         CALL SaveData(NoVar%)
              COLOR ColArr%(15),ColArr%(1)
              LOCATE 20,13,0 : PRINT "Press"
              COLOR ColArr%(12),ColArr%(10)
              LOCATE 20,20,0 : PRINT " ENTER "
              LOCATE 21,20,0 : PRINT " E "
              COLOR ColArr%(15),ColArr%(1)
              LOCATE 20,29,0
17790         IF UCASE$(DescrVar$) = "CONSTANT"_
                 OR UCASE$(DescrVar$) = "DUMMY" THEN
                PRINT "to CONTINUE entering variables "
                LOCATE 21,24,0 : PRINT "to END creation of sub-equation "
              ELSE
                PRINT "to INPUT data for the variable "
                LOCATE 21,24,0 : PRINT ": to END creation of sub-equation ";
                COLOR ColArr%(12),ColArr%(10)
                LOCATE 22,20,0 : PRINT " C ";
                COLOR ColArr%(15),ColArr%(1)
```

```
              PRINT ": to ENTER more variables"
            END IF
17800       DO
              DO : LOOP UNTIL INSTAT
              NewVar$ = UCASE$(INKEY$)
              SELECT CASE NewVar$
                CASE CHR$(13),CHR$(69),CHR$(67)
                  Help$ = "N"
                  EXIT LOOP
              END SELECT
            LOOP
          END IF
        END IF
      LOOP UNTIL Help$ <> "E"
      'enter the data for the non-constant variables, or enter more
      'variables
      IF Help$ <> "Q" THEN
        IF NewVar$ = CHR$(13) THEN
          Neck$ = DescrVar$
          'to enter the data for each variable
          IF UCASE$(DescrVar$) <> "CONSTANT"_
             AND UCASE$(DescrVar$) <> "DUMMY" THEN
            CALL InputData("E")
            IF Help$ <> "Q" THEN
              FOR ValDat% = 1 TO 10
                Values!(ValDat%,NoVar%) = Dat!(ValDat%,2)
              NEXT ValDat%
              'use the smallest number of data
              IF NoVal% > NoData% THEN NoVal% = NoData%
              ERASE Dat!
              SecondVar$ = "Y"  :  LastVar% = NoVar%
              CALL EqScr
            END IF
          END IF
          IF EndVar$ <> "Y" AND Help$ <> "Q" THEN
            COLOR ColArr%(12),ColArr%(10)
            LOCATE 3,45,0  :  PRINT USING " \          \ "; PerformanceM$
            LOCATE 4,45,0  :  PRINT USING " \          \ "; HighwayNo$
            LOCATE 5,45,0  :  PRINT USING " \          \ "; SubEqId$
            COLOR ColArr%(15),ColArr%(1)
          END IF
        ELSEIF NewVar$ = CHR$(69) THEN
          DescrVar$ = ""  :  IDvar$ = ""
          LastVar% = NoVar%
          S% = NoVar% + 1
          FOR NoVar% = S% TO 5
            CALL SaveData(NoVar%)
            FOR RowValues% = 1 TO 10
              Values!(RowValues%,NoVar%) = 0
            NEXT RowValues%
          NEXT NoVar%
          LOCATE 21,10,0  : PRINT SPACE$(60)
```

```
          LOCATE 22,10,0  : PRINT SPACE$(60)
          COLOR ColArr%(15),ColArr%(1)
          FOR I% = 8 TO 18
            LOCATE I%,20,0
            PRINT EqScr$(I%)
          NEXT I%
        ELSEIF NewVar$ = CHR$(67) THEN
          FOR RwVl% = 1 TO 10
            Values!(RwVl%,NoVar%) = 0
          NEXT RwVl%
        END IF
        IF EndVar$ = "Y" OR NewVar$ = CHR$(69) THEN EXIT IF
        LOCATE 21,10,0  : PRINT SPACE$(60)
        LOCATE 22,10,0  : PRINT SPACE$(60)
      END IF
      IF Help$ = "Q" OR EndVar$ = "Y" OR NewVar$ = CHR$(69) THEN EXIT FOR
    NEXT NoVar%
    IF Help$ <> "Q" THEN
      'compute the values of performance for each subequation
      CALL Evaluate(Values!(),Operation!(),SuEqVa!(),NoVal%,EqAddr%())
      'store everything in the data base file
      LSET Performance$ = PerformanceM$
      LSET PerfDesign$ = MKS$(PfDesign!)
      LSET Highway$ = HighwayNo$
      LSET EquationID$ = SubEqId$
      LSET NoVar$ = MKI$(LastVar%)
      LSET PefUnits$ = PfUnit$
      IF EqAddr%(2) < EqAddr%(1) THEN
        LSET PBAindex$ = Null$
        LSET XvUnit$ = UnitsX$
        'save the data
        EqSet& = INT((LOF(4)/472) + 1)
        PUT # 4, EqSet&
      END IF
      ERASE Values!,Operation!,Operation$
      INCR EqAddr%(2)
    ELSE
      ERASE Values!,Operation!,Operation$
    END IF
    IF Help$ = "Q" THEN EXIT LOOP
  LOOP UNTIL EqAddr%(2) > EqAddr%(1)
  IF Help$ <> "Q" THEN
    IF EqAddr%(1) > 1 THEN
      CALL EqRel
      CALL InRel
    END IF
  END IF
  IF Help$ <> "Q" THEN
    'compute the performance for the main equation, based on the results
    'of the subequations
    DIM DYNAMIC PSI!(0:10)
    CALL PerMain(PSI!(),NoVal%)
```

```
        PSI!(0) = PfDesign!
        ERASE SuEqVa!
      END IF
      IF Help$ = "Q" THEN ERASE SuEqVa!
17850 ERASE EqScr$
END SUB
'..............................................................................
'this routine moves the cursor down
SUB VarDown(VarX%)
      INCR VarX%
      LOCATE VarX%,46,1
END SUB
'------------------------------------------------------------------------------
'the following routine computes the performance index, i.e. the area between
'the field performance curve and the design performance value
SUB PfIndex(AreaBetween!)
      SHARED PSI!(),ESALu!(),NoVal%,PfDesign!
      AreaUnder! = 0  :  AreaBetween! = 0
      FOR Area% = 2 TO NoVal%
        AreaUnder! = AreaUnder! + ((PSI!(Area%) + PSI!(Area%-1))/2) *_
                                  (ESALu!(Area%) - ESALu!(Area%-1))
      NEXT Area%
      AreaRectangle! = 0
      FOR Area% = 2 TO NoVal%
        AreaRectangle! = AreaRectangle! + ((ESALu!(Area%) - ESALu!(Area%-1))_
                         * PfDesign!)
      NEXT Area%
      AreaBetween! = ABS(AreaRectangle! - AreaUnder!)
END SUB
'------------------------------------------------------------------------------
'this subroutine is used to compute the performance values
SUB Evaluate(Values!(2),Operation!(2),SuEqVa!(2),NoVal%,EqAddr%(1))
      SHARED Operation$()
      LOCAL DatRow%,DaCol%,Solve!,FunRow%,Operator!,OpRow%
      Answer! = 0
      FOR DaRow% = 1 TO NoVal%
        FOR DaCol% = 1 TO 5
          Solve! = Values!(DaRow%,DaCol%)
          IF Solve! <> 0 THEN
            FOR OpRow% = 9 TO 1 STEP -1
              Operator! = Operation!(OpRow%,DaCol%)
              SELECT CASE Operator!
                CASE 1000000
                  EXIT SELECT
                CASE 2000000
                  Solve! = LOG10(Solve!)
                CASE 3000000
                  Solve! = 10^(Solve!)
                CASE 4000000
                  Solve! = LOG(Solve!)
                CASE 5000000
                  Solve! = EXP(Solve!)
```

```
      CASE ELSE
        SELECT CASE OpRow%
          CASE 1
            IF Operation!(10,DaCol%) = 0 THEN
              Solve! = Operator! * Solve!
            END IF
          CASE 6
            Solve! = Operator! * Solve!
          CASE 7
            Solve! = (Operator!)^(Solve!)
          CASE 8
            Solve! = (Solve!)^(Operator!)
          CASE 9
            Solve! = Solve! + Operator!
        END SELECT
    END SELECT
  NEXT OpRow%
END IF
'use the optional variable relation
SELECT CASE CINT(VAL(Operation$(10,DaCol%)))
  CASE 10
    Answer! = LOG10(Answer!)
  CASE 20
    Answer! = 10^(Answer!)
  CASE 30
    Answer! =  LOG(Answer!)
  CASE 40
    Answer! = EXP(Answer!)
  CASE 50
    Answer! = Answer! * Solve!
  CASE 60
    Answer! = Answer!/Solve!
  CASE 70,0
    Answer! = Answer! + Solve!
  CASE 80
    Answer! = Answer! - Solve!
  CASE 90
    Answer! = Answer! * Operation!(1,DaCol%)
  CASE ELSE
    Code$ = Operation$(10,DaCol%)
    'determine if a decimal figure is being used
    IF MID$(Code$,2,1) <> CHR$(46) THEN
      AddOp$ = MID$(Code$,5,7)
      'number of decimal figures
      DecFi$ = MID$(Code$,3,1)
      DecFi% = VAL(DecFi$)
      'number of integer
      InFi$ = MID$(Code$,2,1)
      InFi% = VAL(InFi$)
      DecFi$ = MID$(AddOp$,InFi%+1,DecFi%)
      InFi$ = MID$(AddOp$,1,InFi%)
      AddOp$ = InFi$ + CHR$(46) + DecFi$
```

```
                AddOp! = VAL(AddOp$)
              ELSE
                NoFig% = VAL(MID$(Code$,3,1))
                AddOp$ = MID$(Code$,4,NoFig%)
                AddOp! = VAL(AddOp$)
              END IF
              SELECT CASE MID$(Code$,1,1)
                CASE CHR$(49)
                  Answer! = (Answer!)^(1/AddOp!)
                CASE CHR$(50)
                  Answer! = (Answer!)^(AddOp!)
                CASE CHR$(51)
                  Answer! = 1/Answer!
                CASE CHR$(52)
                  Answer! = Answer! + Solve! + AddOp!
                CASE CHR$(53)
                  Answer! = Answer! + (Solve! - AddOp!)
                CASE CHR$(54)
                  Answer! = Answer! + AddOp!
                CASE CHR$(55)
                  Answer! = Answer! - AddOp!
              END SELECT
          END SELECT
          Operator! = 0 : AddOp! = 0 : AddOp$ = "" : Code$ = ""
17950   NEXT DaCol%
        SuEqVa!(DaRow%,EqAddr%(2)) = Answer!
        Answer! = 0
      NEXT DaRow%
END SUB
'---------------------------------------------------------------------------
'computes the performance values for the main equation given the values of
'the subequations
SUB PerMain(PSI!(1),NoVal%)
      SHARED SuEqVa!(),EqRel%(),EqAddr%()
      FOR Ini% = 1 TO NoVal%  'get the values of sub-equation 1
        PSI!(Ini%) = SuEqVa!(Ini%,1)
      NEXT Ini%
      SuEq% = 2
      DO
        FOR PV% = 1 TO NoVal%
          SELECT CASE EqRel%(SuEq% - 1)
            CASE 9
              PSI!(PV%) = PSI!(PV%) + SuEqVa!(PV%,SuEq%)
            CASE 10
              PSI!(PV%) = PSI!(PV%) - SuEqVa!(PV%,SuEq%)
            CASE 11
              PSI!(PV%) = PSI!(PV%) * SuEqVa!(PV%,SuEq%)
            CASE 12
              PSI!(PV%) = PSI!(PV%) / SuEqVa!(PV%,SuEq%)
            CASE 13
              PSI!(PV%) = PSI!(PV%)^(SuEqVa!(PV%,SuEq%))
            CASE 14
```

```
          PSI!(PV%) = EXP(PSI!(PV%))
        CASE 15
          IF PSI!(PV%) <> 0 THEN PSI!(PV%) = LOG(PSI!(PV%))
        CASE 16
          IF PSI!(PV%) <> 0 THEN PSI!(PV%) = LOG10(PSI!(PV%))
        CASE 17
          PSI!(PV%) = EXP10(PSI!(PV%))
      END SELECT
    NEXT PV%
    INCR SuEq%
  LOOP UNTIL SuEq% > EqAddr%(1)
  ERASE EqRel%
END SUB
'-----------------------------------------------------------------------------
'store the data in the data base file
SUB SaveData(NoVar%)
  SHARED VarDesc1$,VarCoef1$,LogVar1$,Var101$,LnVar1$,eVar1$,ConsVar1$
  SHARED RootVar1$,ExpVar1$,VarConst1$,VarOp1$,VarDesc2$,VarCoef2$,LogVar2$
  SHARED Var102$,LnVar2$,eVar2$,ConsVar2$,RootVar2$,ExpVar2$,VarConst2$
  SHARED VarOp2$,VarDesc3$,VarCoef3$,LogVar3$,Var103$,LnVar3$,eVar3$
  SHARED ConsVar3$,RootVar3$,ExpVar3$,VarConst3$,VarOp3$,VarDesc4$,VarCoef4$
  SHARED LogVar4$,Var104$,LnVar4$,eVar4$,ConsVar4$,RootVar4$,ExpVar4$
  SHARED VarConst4$,VarOp4$,VarDesc5$,VarCoef5$,LogVar5$,Var105$,LnVar5$
  SHARED eVar5$,ConsVar5$,RootVar5$,ExpVar5$,VarConst5$,VarOp5$
  SHARED DescrVar$,Operation!(),Operation$()
  SELECT CASE NoVar%
    CASE 1
      LSET VarDesc1$ = DescrVar$
      LSET VarCoef1$ = MKS$(Operation!(1,1))
      LSET LogVar1$  = MKS$(Operation!(2,1))
      LSET Var101$ = MKS$(Operation!(3,1))
      LSET LnVar1$  = MKS$(Operation!(4,1))
      LSET eVar1$  = MKS$(Operation!(5,1))
      LSET ConsVar1$ = MKS$(Operation!(6,1))
      LSET RootVar1$ = MKS$(Operation!(7,1))
      LSET ExpVar1$ = MKS$(Operation!(8,1))
      LSET VarConst1$ = MKS$(Operation!(9,1))
      LSET VarOp1$ = Operation$(10,1)
    CASE 2
      LSET VarDesc2$ = DescrVar$
      LSET VarCoef2$ = MKS$(Operation!(1,2))
      LSET LogVar2$  = MKS$(Operation!(2,2))
      LSET Var102$ = MKS$(Operation!(3,2))
      LSET LnVar2$  = MKS$(Operation!(4,2))
      LSET eVar2$  = MKS$(Operation!(5,2))
      LSET ConsVar2$ = MKS$(Operation!(6,2))
      LSET RootVar2$ = MKS$(Operation!(7,2))
      LSET ExpVar2$ = MKS$(Operation!(8,2))
      LSET VarConst2$ = MKS$(Operation!(9,2))
      LSET VarOp2$ = Operation$(10,2)
    CASE 3
      LSET VarDesc3$ = DescrVar$
```

```
      LSET VarCoef3$ = MKS$(Operation!(1,3))
      LSET LogVar3$  = MKS$(Operation!(2,3))
      LSET Var103$ = MKS$(Operation!(3,3))
      LSET LnVar3$  = MKS$(Operation!(4,3))
      LSET eVar3$  = MKS$(Operation!(5,3))
      LSET ConsVar3$ = MKS$(Operation!(6,3))
      LSET RootVar3$ = MKS$(Operation!(7,3))
      LSET ExpVar3$ = MKS$(Operation!(8,3))
      LSET VarConst3$ = MKS$(Operation!(9,3))
      LSET VarOp3$ = Operation$(10,3)
    CASE 4
      LSET VarDesc4$ = DescrVar$
      LSET VarCoef4$ = MKS$(Operation!(1,4))
      LSET LogVar4$  = MKS$(Operation!(2,4))
      LSET Var104$ = MKS$(Operation!(3,4))
      LSET LnVar4$  = MKS$(Operation!(4,4))
      LSET eVar4$  = MKS$(Operation!(5,4))
      LSET ConsVar4$ = MKS$(Operation!(6,4))
      LSET RootVar4$ = MKS$(Operation!(7,4))
      LSET ExpVar4$ = MKS$(Operation!(8,4))
      LSET VarConst4$ = MKS$(Operation!(9,4))
      LSET VarOp4$ = Operation$(10,4)
    CASE 5
      LSET VarDesc5$ = DescrVar$
      LSET VarCoef5$ = MKS$(Operation!(1,5))
      LSET LogVar5$  = MKS$(Operation!(2,5))
      LSET Var105$ = MKS$(Operation!(3,5))
      LSET LnVar5$  = MKS$(Operation!(4,5))
      LSET eVar5$  = MKS$(Operation!(5,5))
      LSET ConsVar5$ = MKS$(Operation!(6,5))
      LSET RootVar5$ = MKS$(Operation!(7,5))
      LSET ExpVar5$ = MKS$(Operation!(8,5))
      LSET VarConst5$ = MKS$(Operation!(9,5))
      LSET VarOp5$ = Operation$(10,5)
  END SELECT
END SUB
'-----------------------------------------------------------------------------
'this routine is used to input the relationships between sub-equations
SUB InRel
      SHARED EqRel%(),ColArr%(),EqAddr%(),Rel$(),Help$,OnHelp$
      DIM DYNAMIC EqRel%(1:4)
      RelR% = 9  :  OnHelp$ = "RELATION"
      C% = 35
      RelCou% = 1
      DO
17960   COLOR ColArr%(12),ColArr%(10)
        LOCATE 9,26,0 : PRINT CHR$(0) + Rel$(9) + SPACE$(8-LEN(Rel$(9)))
        LOCATE 9,C%,0 : PRINT " x "
        DO
17970     DO  :  LOOP UNTIL INSTAT
          EqR$ = INKEY$
          SELECT CASE EqR$
```

```
          CASE CHR$(13)
            EqRel%(RelCou%) = RelR%
            RelCou% = RelCou% + 1
            IF RelCou% < EqAddr%(1) THEN
              LOCATE RelR%,C%,0   :   PRINT " x "
              C% = C% + 4
              COLOR ColArr%(15),ColArr%(1)
              LOCATE RelR%,26,0
              PRINT CHR$(0) + Rel$(RelR%) + SPACE$(8-LEN(Rel$(RelR%)))
              RelR% = 9
              EXIT LOOP
            END IF
          CASE CHR$(27)
            Help$ = "Q"
            ERASE EqRel%
          CASE ELSE
            SELECT CASE MID$(EqR$,2,1)
              CASE CHR$(59)
                CALL PBAhelp(OnHelp$)
              CASE CHR$(60)
                A% = 30
                COLOR ColArr%(12),ColArr%(15)
                FOR Label% = 65 TO (63 + EqAddr%(1))
                  FOR Rel% = 9 TO 17
                    LOCATE Rel%,(Label%-A%),0
                    PRINT SPACE$(3)
                  NEXT Rel%
                  A% = A% - 3
                NEXT Label%
                RelR% = 9   :   C% = 35   :   RelCou% = 1
                EXIT LOOP
              CASE CHR$(72),CHR$(80)
                CALL RelD(RelR%,C%)
            END SELECT
        END SELECT
        IF RelCou% >= EqAddr%(1) OR Help$ = "Q" THEN EXIT LOOP
      LOOP
      IF Help$ = "Q" OR RelCou% >= EqAddr%(1) THEN EXIT LOOP
    LOOP
    ERASE Rel$
END SUB
'..........................................................................
'print the labels for the relationships
SUB RelD(RelR%,C%)
      SHARED ColArr%(),Rel$()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE RelR%,26,0
      PRINT CHR$(0) + Rel$(RelR%) + SPACE$(8-LEN(Rel$(RelR%)));
      COLOR ColArr%(12),ColArr%(15)
      LOCATE RelR%,C%,0   :   PRINT "   "
      RelR% = RelR% + 1
      IF RelR% = 18 THEN RelR% = 9
```

```
      COLOR ColArr%(12),ColArr%(10)
      LOCATE RelR%,26,0
      PRINT CHR$(0) + Rel$(RelR%) + SPACE$(8-LEN(Rel$(RelR%)))
      LOCATE RelR%,C%,0  :  PRINT " x "
END SUB
'-----------------------------------------------------------------------------
'create the screen for entering the performance equation
SUB EqScr
      SHARED ColArr%(),SecondVar$,PerformanceM$,HighwayNo$,SubEqId$,VarId$
      SHARED DescrVar$,NoVar%,EndVar$,EqScr$(),Values!()
      CALL Frame
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 2,22,0  :  PRINT " User's Performance Equation "
      LOCATE 24,7,0  :  PRINT " F1 : Help ";
      LOCATE 24,19,0 :  PRINT " F2 : Edit ";
      LOCATE 24,31,0 :  PRINT " ESC : Quit ";
      LOCATE 24,44,0 :  PRINT " " + CHR$(24) + CHR$(25) + " : Up, down ";
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 3,20,0  :  PRINT "Performance Measure  :"
      LOCATE 4,20,0  :  PRINT "Pavement Section #   :"
      LOCATE 5,20,0  :  PRINT "Equation ID number   :"
      LOCATE 6,20,0  :  PRINT "Variable ID number   :"

      IF SecondVar$ = "Y" THEN
        SecondVar$ = ""
        COLOR ColArr%(12),ColArr%(10)
        LOCATE 3,45,0  :  PRINT USING " \           \ "; PerformanceM$
        LOCATE 4,45,0  :  PRINT USING " \           \ "; HighwayNo$
        LOCATE 5,45,0  :  PRINT USING " \           \ "; SubEqId$
        LOCATE 6,45,0  :  PRINT USING "  &  "; VarId$
        LOCATE 20,20,0 :  PRINT " ENTER "
        LOCATE 21,20,0 :  PRINT " E "
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 20,13,0 :  PRINT "Press"
        LOCATE 21,13,0 :  PRINT "Press"
        LOCATE 20,29,0 :  PRINT "to CONTINUE entering variables "
        LOCATE 21,24,0 :  PRINT "to END creation of equation "
        DO
          DO  :  LOOP UNTIL INSTAT
          EndVar$ = UCASE$(INKEY$)
          SELECT CASE EndVar$
            CASE CHR$(69)
              DescrVar$ = ""
              S% = NoVar% + 1
              FOR NoVar% = S% TO 5
                CALL SaveData(NoVar%)
                FOR RowVal% = 1 TO 10
                  Values!(RowVal%,NoVar%) = 0
                NEXT RowVal%
              NEXT NoVar%
              EndVar$ = "Y"
              LOCATE 20,10,0  :  PRINT SPACE$(65)
```

```
            LOCATE 21,10,0  :  PRINT SPACE$(65)
            EXIT LOOP
          CASE CHR$(13)
            EXIT LOOP
        END SELECT
      LOOP
    END IF
    FOR I% = 8 TO 19
      LOCATE I%,20,0
      PRINT EqScr$(I%)
    NEXT I%
END SUB
'---------------------------------------------------------------------------------
'this routine creates the screen for entering the relationships between the
'different subequations
SUB EqRel
39000 SHARED ColArr%(),HighwayNo$,EqID$,PerformanceM$,EqAddr%(),Rel$()
      CALL Frame
      COLOR ColArr%(12),ColArr%(10)
      LOCATE 2,14,0
      PRINT " Relationships between Performance Sub-Equations "
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 3,27,0 : PRINT USING " Highway ID  : &"; LEFT$(HighwayNo$,10)
      LOCATE 4,27,0 : PRINT USING " Equation ID : &"; LEFT$(EqID$,10)
      LOCATE 5,27,0 : PRINT USING " Performance : &"; PerformanceM$;
      PRINT SPACE$(10-LEN(PerformanceM$))
      LOCATE 7,35,0 : PRINT " Sub-Equations "
      DIM DYNAMIC Label$(65:68)
       Label$(65) = " A : 1-2 "
       Label$(66) = " B : A-3 "
       Label$(67) = " C : B-4 "
       Label$(68) = " D : C-5 "
      A% = 30  :  B% = 20
39100 FOR Label% = 65 TO (63 + EqAddr%(1))
        LOCATE 8,(Label%-A%),0
        PRINT " " + CHR$(Label%) + " "
        FOR Rel% = 9 TO 17
          LOCATE Rel%,(Label%-A%),0
          PRINT SPACE$(3)
        NEXT Rel%
        LOCATE 20,B%,0
        PRINT Label$(Label%)
        A% = A% - 3  :  B% = B% + 10
      NEXT Label%
      ERASE Label$
39150 LOCATE 24,4,0  :  PRINT " " + CHR$(24) + CHR$(25) + " : Vertical ";
      LOCATE 24,20,0 :  PRINT " ENTER : relation ";
      LOCATE 24,39,0 :  PRINT " F1 : Help ";
      LOCATE 24,51,0 :  PRINT " F2 : Edit ";
      LOCATE 24,63,0 :  PRINT " ESC : Quit ";
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 8,25,0  :  PRINT "Relation"
```

```
     DIM DYNAMIC Rel$(9:17)
      Rel$(9)  = CHR$(43)
      Rel$(10) = CHR$(45)
      Rel$(11) = CHR$(42)
      Rel$(12) = CHR$(47)
      Rel$(13) = "Exp"
      Rel$(14) = "e^"
      Rel$(15) = "Ln"
      Rel$(16) = "Log"
      Rel$(17) = "10^"
     FOR Rp% = 9 TO 17
       LOCATE Rp%,27,0
       PRINT Rel$(Rp%)
     NEXT Rp%
     LOCATE 19,20,0 :  PRINT "Relationship between:"
END SUB
'----------------------------------------------------------------------------
'see the contents of an equation file
SUB ViewEqn(FileEqn$)
     SHARED VarDesc1$,VarCoef1$,LogVar1$,Var101$,LnVar1$,eVar1$,ConsVar1$
     SHARED RootVar1$,ExpVar1$,VarConst1$,VarOp1$,VarDesc2$,VarCoef2$,LogVar2$
     SHARED Var102$,LnVar2$,eVar2$,ConsVar2$,RootVar2$,ExpVar2$,VarConst2$
     SHARED VarOp2$,VarDesc3$,VarCoef3$,LogVar3$,Var103$,LnVar3$,eVar3$
     SHARED ConsVar3$,RootVar3$,ExpVar3$,VarConst3$,VarOp3$,VarDesc4$,VarCoef4$
     SHARED LogVar4$,Var104$,LnVar4$,eVar4$,ConsVar4$,RootVar4$,ExpVar4$
     SHARED VarConst4$,VarOp4$,VarDesc5$,VarCoef5$,LogVar5$,Var105$,LnVar5$
     SHARED eVar5$,ConsVar5$,RootVar5$,ExpVar5$,VarConst5$,VarOp5$,OnField$
     SHARED Performance$,Highway$,EquationID$,PerfDesign$,NoVar$,PBAindex$
     SHARED PefUnits$,XvUnit$,ColArr%()
39200 CALL EqnScr(FileEqn$)
     IF OnField$ = "N" THEN
       Section$ = "Highway ID:"
     ELSE
       Section$ = "Pavement ID:"
     END IF
     RecNo& = LOF(4)/472
39220 FOR ViewVar& = 1 TO RecNo&
       GET # 4, ViewVar&
       COLOR ColArr%(15),ColArr%(1)
       LOCATE 3,5,0  : PRINT USING "Record No.: ####"; ViewVar&
       LOCATE 3,40,0
       IF LEFT$(PBAindex$,4) = STRING$(4,48) THEN
         PRINT USING "PBA Index: &"; PBAindex$
       ELSE
         PRINT USING "PBA Index: +#########.#######"; CVS(PBAindex$)
       END IF
       LOCATE 4,5,0
       PRINT USING "Performance Measure: &"; LEFT$(Performance$,15)
       LOCATE 4,40,0 : PRINT USING "& &"; Section$,LEFT$(Highway$,10)
39240  LOCATE 5,5,0  : PRINT USING "Equation ID: &"; LEFT$(EquationID$,12)
       LOCATE 5,40,0
       PRINT USING "Performance Design Value: ###.###"; CVS(PerfDesign$)
```

```
        LOCATE 6,5,0  : PRINT USING "Perf. Units: &"; PefUnits$
        LOCATE 6,40,0 : PRINT USING "X var Units: &"; XvUnit$
        EqnFormat$ = " #######.##### #######.##### #######.##### #######.#####"
        EqnFormat$ = EqnFormat$ + " #######.#####"
        VarFormat$ = " \           \ \           \ \           \ \           \"
        VarFormat$ = VarFormat$ + " \           \"
        LOCATE 8,15,0
        PRINT "Var 1          Var 2          Var 3          Var 4          Var 5"
        LOCATE 9,2,0 : PRINT "Var Coe ";
        PRINT USING EqnFormat$; CVS(VarCoef1$),CVS(VarCoef2$),CVS(VarCoef3$),_
                                CVS(VarCoef4$),CVS(VarCoef5$)
        LOCATE 10,2,0 : PRINT "Log Var ";
        PRINT USING EqnFormat$; CVS(LogVar1$),CVS(LogVar2$),CVS(LogVar3$),_
                                CVS(LogVar4$),CVS(LogVar5$)
        LOCATE 11,2,0 : PRINT "10 Var  ";
        PRINT USING EqnFormat$; CVS(Var101$),CVS(Var102$),CVS(Var103$),_
                                CVS(Var104$),CVS(Var105$)
        LOCATE 12,2,0 : PRINT "Ln Var  ";
        PRINT USING EqnFormat$; CVS(LnVar1$),CVS(LnVar2$),CVS(LnVar3$),_
                                CVS(LnVar4$),CVS(LnVar5$)
        LOCATE 13,2,0 : PRINT "e Var   ";
        PRINT USING EqnFormat$; CVS(eVar1$),CVS(eVar2$),CVS(eVar3$),CVS(eVar4$),_
                                CVS(eVar5$)
        LOCATE 14,2,0 : PRINT "(C) Var ";
        PRINT USING EqnFormat$; CVS(ConsVar1$),CVS(ConsVar2$),CVS(ConsVar3$),_
                                CVS(ConsVar4$),CVS(ConsVar5$)
        LOCATE 15,2,0 : PRINT "Exp Var ";
        PRINT USING EqnFormat$; CVS(ExpVar1$),CVS(ExpVar2$),CVS(ExpVar3$),_
                                CVS(ExpVar4$),CVS(ExpVar5$)
        LOCATE 16,2,0 : PRINT "Root Va ";
        PRINT USING EqnFormat$; CVS(RootVar1$),CVS(RootVar2$),CVS(RootVar3$),_
                                CVS(RootVar4$),CVS(RootVar5$)
        LOCATE 17,2,0 : PRINT "(C)p(V) ";
        PRINT USING EqnFormat$; CVS(VarConst1$),CVS(VarConst2$),_
                                CVS(VarConst3$),CVS(VarConst4$),CVS(VarConst5$)
        LOCATE 18,2,0 : PRINT "V Rel   ";
        PRINT USING VarFormat$; VarOp1$,VarOp2$,VarOp3$,VarOp4$,VarOp5$
        LOCATE 20,3,0  : PRINT USING "Var. 1: &"; VarDesc1$
        LOCATE 20,35,0 : PRINT USING "Var. 2: &"; VarDesc2$
        LOCATE 21,3,0  : PRINT USING "Var. 3: &"; VarDesc3$
        LOCATE 21,35,0 : PRINT USING "Var. 4: &"; VarDesc4$
        LOCATE 22,3,0  : PRINT USING "Var. 5: &"; VarDesc5$
        LOCATE 22,35,0 : PRINT USING "No. Var.: ##"; CVI(NoVar$)
        DO  :  LOOP UNTIL INSTAT
        MoreEqn$ = INKEY$
        CALL ClScr(FileEqn$)
        IF MoreEqn$ = CHR$(27) THEN EXIT FOR
     NEXT ViewVar&
END SUB
'-------------------------------------------------------------------------
' define the length for each field on each record a record represents a
' determined performance equation
```

```
SUB OpenFile(FileEqn$)
      SHARED VarDesc1$,VarCoef1$,LogVar1$,Var101$,LnVar1$,eVar1$,ConsVar1$
      SHARED RootVar1$,ExpVar1$,VarConst1$,VarOp1$,VarDesc2$,VarCoef2$,LogVar2$
      SHARED Var102$,LnVar2$,eVar2$,ConsVar2$,RootVar2$,ExpVar2$,VarConst2$
      SHARED VarOp2$,VarDesc3$,VarCoef3$,LogVar3$,Var103$,LnVar3$,eVar3$
      SHARED ConsVar3$,RootVar3$,ExpVar3$,VarConst3$,VarOp3$,VarDesc4$,VarCoef4$
      SHARED LogVar4$,Var104$,LnVar4$,eVar4$,ConsVar4$,RootVar4$,ExpVar4$
      SHARED VarConst4$,VarOp4$,VarDesc5$,VarCoef5$,LogVar5$,Var105$,LnVar5$
      SHARED eVar5$,ConsVar5$,RootVar5$,ExpVar5$,VarConst5$,VarOp5$
      SHARED Performance$,Highway$,EquationID$,PerfDesign$,NoVar$,PBAindex$
      SHARED PefUnits$,XvUnit$
      OPEN FileEqn$ AS # 4 LEN = 472
      FIELD # 4, 25 AS Performance$, 25 AS Highway$, 25 AS EquationID$,_
                 4 AS PerfDesign$, 2 AS NoVar$, 4 AS PBAindex$,_
                 15 AS PefUnits$, 15 AS XvUnit$,_
                 25 AS VarDesc1$, 4 AS VarCoef1$,_
                 4 AS LogVar1$, 4 AS Var101$, 4 AS LnVar1$, 4 AS eVar1$,_
                 4 AS ExpVar1$, 4 AS ConsVar1$, 4 AS RootVar1$,_
                 4 AS VarConst1$, 10 AS VarOp1$,_
                 25 AS VarDesc2$, 4 AS VarCoef2$,_
                 4 AS LogVar2$, 4 AS Var102$, 4 AS LnVar2$, 4 AS eVar2$,_
                 4 AS ExpVar2$, 4 AS ConsVar2$, 4 AS RootVar2$,_
                 4 AS VarConst2$, 10 AS VarOp2$,_
                 25 AS VarDesc3$, 4 AS VarCoef3$,_
                 4 AS LogVar3$, 4 AS Var103$, 4 AS LnVar3$, 4 AS eVar3$,_
                 4 AS ExpVar3$, 4 AS ConsVar3$, 4 AS RootVar3$,_
                 4 AS VarConst3$, 10 AS VarOp3$,_
                 25 AS VarDesc4$, 4 AS VarCoef4$,_
                 4 AS LogVar4$, 4 AS Var104$, 4 AS LnVar4$, 4 AS eVar4$,_
                 4 AS ExpVar4$, 4 AS ConsVar4$, 4 AS RootVar4$,_
                 4 AS VarConst4$, 10 AS VarOp4$,_
                 25 AS VarDesc5$, 4 AS VarCoef5$,_
                 4 AS LogVar5$, 4 AS Var105$, 4 AS LnVar5$, 4 AS eVar5$,_
                 4 AS ExpVar5$, 4 AS ConsVar5$, 4 AS RootVar5$,_
                 4 AS VarConst5$, 10 AS VarOp5$
END SUB
'--------------------------------------------------------------------------
```

Appendix A.6
PBAASHTO.BAS

```
'
'                              PBAASHTO.BAS
'
'                "Predicted Serviceability Module"
'The following program was coded using Turbo BASIC version 1.1
'This module is to be used for the determination of the amount of
'deterioration due to load-related factors solely.  Refer to section 3.3
'of report, "Effects of Environmental Factors on Pavement Deterioration".
'This module solves for equation 1.2.1, Part I;  and, equation PP.26, Appendix
'PP, of the 1986 AASHTO Guide for the Design of Pavement Structures, and for
'a customized performance model.
'---------------------------------------------------------------------------
'a file to store the information needed is to be created next
'a random access file is to be used as the data base
    $EVENT OFF
    'get the graphics type
    OPEN "PBAVIDEO.DAT" FOR INPUT AS # 3
    INPUT # 3, VideoGraph%
    CLOSE # 3
    'determine the type of monitor
    CALL Monitor(VideoMode%)
    DO
      ON ERROR GOTO ErrorTrap
      'create the menu for file creation
      DO
470     CALL FileScreen
475     NewName$ = ""   : Help$ = "" : OnHelp$ = "IData"
        LOCATE 4,31,0   : PRINT " Data File "
        LOCATE 6,15,0   : PRINT "1. Create a new data file"
        LOCATE 7,15,0   : PRINT "2. Open an existing data file"
        LOCATE 8,15,0   : PRINT "3. View a data file"
        LOCATE 9,15,0   : PRINT "4. Return to PBA Main Menu"
        LOCATE 10,15,0 : PRINT " Enter a choice : "
        'select an option
        DO
          CALL MenuChoice(Choice$,10,36)
          SELECT CASE Choice$
            CASE CHR$(49),CHR$(50),CHR$(51),CHR$(52)
              EXIT LOOP
          END SELECT
        LOOP
        FileChoice% = VAL(Choice$)    'to check existence of file in PBAFILE.BAS
        Viewing$ = "N"    'flag for FiNaCheck
485     SELECT CASE Choice$
          CASE CHR$(49)    ' to create a brand new data file
            LOCATE 4,30,0   :  PRINT " New Data File "
            FileSpec$ = " created"
          CASE CHR$(50)    ' to open an existing data file to add or modify
            LOCATE 4,30,0   :  PRINT " Open Data File "
            FileSpec$ = " opened"
          CASE CHR$(51)
            LOCATE 4,30,0   :  PRINT " View Data File "
```

```
          FileSpec$ = " seen"  :  Viewing$ = "Y"
        CASE CHR$(52)
          COLOR ColArr%(15),ColArr%(1)
          CLS
          SYSTEM
      END SELECT
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 6,13,0    ' screen for entering the name of the file
      PRINT "Enter the name of the data file to be"; : PRINT FileSpec$
      'enter the name of the file and check its existence
      FileExt$ = ".DAT"
      CALL FiNaCheck(FileExt$,FileChoice%)
      FileMain$ = ""
    LOOP UNTIL NewName$ <> "Y"
    'store the name of the user's data file
    LOCATE 9,15,0
    OPEN "FILENAME.PBA" FOR OUTPUT AS #3
    PRINT # 3, USING "  ###  & "; Drive%,FileErr$
    PRINT # 3, USING "## ##"; VideoGraph%
    CLOSE
    SELECT CASE FileChoice%
      CASE 1,2
        ERASE ColArr%
        GOSUB DataIn
        CALL Monitor(VideoMode%)
      CASE 3
        WorkFile$ = FileErr$
        GOSUB OpenData
        GOSUB ViewFile
        CLOSE # 1
    END SELECT
  LOOP
'------------------------------------------------------------------------
'the following routine is the data collection routine
DataIn:
    DO
      NoSpace$ = "" : OnHelp$ = "IData"
750   CALL Monitor(VideoMode%)
      CALL Border("N")
      OPEN "FILENAME.PBA" FOR INPUT AS # 3
      INPUT # 3, Drive%,WorkFile$
      INPUT # 3, VideoGraph%
      CLOSE # 3
'determine if at least 1K bytes is available in the default or indicated disk
      NofRecords& = INT(FNbytes(Drive%) / 197) - 6
755   IF NofRecords& < 5 THEN ERROR 61
      IF NoSpace$ <> "Y" THEN
        'define data records
        GOSUB OpenData
'create an array to store the headlines for inputing the data
        DIM DYNAMIC dc$(8:16)
         dc$(8) = "Highway #"
```

```
        dc$(9) = "District"
        dc$(10) = "County"
        dc$(11) = "Begin M.P."
        dc$(12) = "End M.P."
        dc$(13) = "Section EA"
        dc$(14) = "EA/SJ"
        dc$(15) = "Year"
        dc$(16) = "Region"
'the Identification$ array stores the input data before storing them in dato$
        DIM DYNAMIC Identification$(8:16)
'call the data collection screen
'the row variable is used to return to the row after calling the help screen
'and for guidance in storing the input data in the different arrays
        Editing$ = "I"   'ID for editing routine
        RowID% = 8
        WHILE RowID% <= 16
          COLOR ColArr%(12),ColArr%(15)
1000      LOCATE RowID%,26,0
          PRINT USING "    \              \:" + SPACE$(18); dc$(RowID%);
          RowDat% = RowID%
          CALL InputString(StringIN$,RowDat%,51,13)
          IF Help$ <> "Q" THEN
            Identification$(RowID%) = StringIN$
            LOCATE RowID%,26,0
            COLOR ColArr%(15),ColArr%(1)
            PRINT USING "    \              \:"; dc$(RowID%);
            INCR RowID%
          ELSE
            EXIT LOOP
          END IF
        WEND
        IF Help$ <> "Q" THEN
1090      DO
            COLOR ColArr%(15),ColArr%(1)
            LOCATE 18,15,0  : PRINT "Performance Measure:" + SPACE$(25)
            LOCATE 19,14,0
            PRINT " 1. User's Performance, AASHTO-PSI" + SPACE$(25)
            LOCATE 20,14,0  : PRINT " 2. AASHTO-Deflection" + SPACE$(25)
            LOCATE 21,25,0  : PRINT "Enter a choice :"
            m$ = ""  :  p$ = ""  :  Continue$ = ""   'clean the variables
            DO
              CALL MenuChoice(Choice$,21,43)
              SELECT CASE Choice$
                CASE CHR$(49),CHR$(50)
                  EXIT LOOP
                CASE CHR$(27)
                  Help$ = "Q"
              END SELECT
            LOOP UNTIL Help$ = "Q"
            'enter traffic data
            IF Help$ <> "Q" THEN
              PefCho% = VAL(Choice$)
```

```
                COLOR ColArr%(15),ColArr%(1)
                SELECT CASE PefCho%
                  CASE 1
                    OnDefl$ = "N"
                    'move to the traffic data
                    LOCATE 18,15,0
                    PRINT "Independent Performance Variable (X axis):"
1620                LOCATE 19,14,0
                    PRINT " Predicted Traffic (y/n) : " + SPACE$(25)
                    LOCATE 20,14,0
                    PRINT " Manual Input (y/n) : " + SPACE$(25)
                    COLOR ColArr%(12),ColArr%(15)
1630                LOCATE 19,14,0
                    PRINT " Predicted Traffic (y/n) :" + SPACE$(4)
1650                RowID% = 19
                    DO
                      CALL MenuChoice(Choice$,19,42)
                      SELECT CASE Choice$
                        CASE CHR$(27)
                          Help$ = "Q"
                        CASE "Y","y","N","n"
                          EXIT LOOP
                      END SELECT
                    LOOP UNTIL Help$ = "Q"
                    IF Help$ <> "Q" THEN
                      p$ = UCASE$(Choice$)
1675                  DO
                        Help$ = ""  :  Cht$ = "N"
                        SELECT CASE p$
                          CASE "Y"
                            GOSUB PredictedTraffic
                            IF Help$ = "Q" THEN
                              IF ChT$ = "Y" THEN
                                ERASE Vehicle$
                                m$ = "Y" : p$ = "N"
                                EXIT SELECT    'GOTO 1750
                              END IF
                            END IF
                            Continue$ = "Y"  :  EXIT LOOP
                          CASE "N"
                            IF m$ <> "Y" THEN
                              LOCATE 19,14,0 : COLOR ColArr%(15),ColArr%(1)
                              PRINT " Predicted Traffic (y/n) : "
                              COLOR ColArr%(12),ColArr%(15)
1720                          LOCATE 20,14,0
                              PRINT " Manual Input (y/n) :" + SPACE$(4)
1740                          RowID% = 20
                              DO
                                CALL MenuChoice(Choice$,20,37)
                                SELECT CASE Choice$
                                  CASE "Y","y","N","n"
                                    EXIT LOOP
```

```
                               END SELECT
                             LOOP
                             m$ = UCASE$(Choice$)
                           END IF
1750                       SELECT CASE m$
                             CASE "Y"
                               CALL ManTraffic
                               IF Help$ = "Q" THEN
                                 IF ChT$ = "Y" THEN p$ = "Y"
                                 EXIT SELECT   'GOTO 1675
                               END IF
                               IF opractico$ = "yes" THEN opractico$ = ""
                               Continue$ = "Y"
                               EXIT LOOP
                             CASE "N"
                               EXIT LOOP
                           END SELECT
                       END SELECT
                     LOOP UNTIL ChT$ <> "Y"
                     IF Continue$ = "Y" THEN EXIT LOOP
                   END IF
                 CASE 2
                   OnDefl$ = "Y"
                   EXIT LOOP
               END SELECT
             END IF
           LOOP UNTIL Help$ = "Q"
           IF Help$ <> "Q" THEN
             'store the input data in the dato$ file
1920         LSET highwaynumber$ = Identification$(8)
             LSET district$ = Identification$(9)
             LSET county$ = Identification$(10)
             LSET beginMP$ = MKS$(VAL(Identification$(11)))
             LSET endMP$ = MKS$(VAL(Identification$(12)))
             LSET sectionEA$ = Identification$(13)
             LSET easj$ = Identification$(14)
             LSET year$ = Identification$(15)
             LSET region$ = Identification$(16)
             GOSUB PerfEqn
           ELSE
             CLOSE
             ERASE Identification$,dc$,ColArr%
           END IF
         ELSE
           CLOSE
           ERASE Identification$,dc$,ColArr%
         END IF
       END IF
     LOOP UNTIL Help$ = "Q"
RETURN
'------------------------------------------------------------------------------
'define the fields
```

```
OpenData:
        OPEN WorkFile$ AS # 1 LEN = 197
'define the length for each field on each record, a record represents a
'determined pavement section
        FIELD #1, 25 AS highwaynumber$, 25 AS district$, 25 AS county$,_
                  4 AS beginMP$, 4 AS endMP$, 25 AS sectionEA$, 25 AS easj$,_
                  4 AS year$, 4 AS region$, 8 AS modulus$, 4 AS deviate$,_
                  4 AS deviation$, 4 AS StructuralNumber$, 4 AS PBAindex$,_
                  4 AS DynamicLoad$, 4 AS PlateRadius$, 4 AS PavementThickness$,_
                  4 AS SubgradePoisson$, 4 AS FieldSN$, 4 AS DeflectionFactor$,_
                  4 AS DeflSN$, 4 AS DeflDesign$
RETURN
'---------------------------------------------------------------------------
'the following routine is used to determine the type of performance equation
'the user is going to use
PerfEqn:
        DO
1922      NewEq$ = "" : PerfEqn$ = "Y" : User$ = "" : Dscr$ = "N"    'flags
          CALL PerfScr
          PerfEqn$ = ""  :  OnHelp$ = "IData"
          LOCATE 3,30,0  :  PRINT "Performance Equation"
          LOCATE 5,12,0
          PRINT "1. AASHTO Equations for Flexible Pavements."
          LOCATE 6,12,0
          PRINT "   Equations 1.2.1 & PP.26 of 1986 AASHTO Design Guide."
          LOCATE 7,12,0
          PRINT "2. User-Defined Performance Equation"
          LOCATE 8,12,0
          PRINT "3. Cancel data for current section"
          LOCATE 10,12,0  :  PRINT "Enter a choice : "
          DO
            CALL MenuChoice(Choice$,10,30)
            PerChoice% = VAL(Choice$)
            SELECT CASE Choice$
              CASE CHR$(49)             'use AASHTO equations
                'to store the data to be plotted
                DIM DYNAMIC Xval!(1:10),Yval!(1:10)
                User$ = "N"  :  PerformanceM$ = "PSI"  :  PfDesign! = 4.2
                Ymax! = 5  :  Ymin! = 0  : Xvar$ = "ESAL (* 1E6)"
                GOSUB AASHTO
                'if quiting
                IF Help$ = "Q" THEN
                  ERASE Xval!,Yval!
                END IF
              CASE CHR$(50)             'user-defined equations
                'to store the data to be plotted
                DIM DYNAMIC Xval!(1:10),Yval!(1:10)
                CALL UserEqn("N")
                IF Help$ = "C" THEN
                  ERASE Xval!,Yval!
                ELSE
                  CALL PefWait("Y")
```

```
                User$ = "Y"
                CALL PBAindex
                Help$ = "S"
                'clean the memory to be saved
                DIM DYNAMIC Deflection!(17:22)
                deflection! = 0
                CALL SaveDfl
                ERASE Deflection!
                ResilientModulus# = 0
                LSET modulus$ = MKD$(ResilientModulus#)
                StandDev! = 0 : LSET deviate$ = MKS$(StandDev!)
                StandardDeviation! = 0
                LSET deviation$ = MKS$(StandardDeviation!)
                StructuralNumber! = 0
                LSET StructuralNumber$ = MKS$(StructuralNumber!)
              END IF
              OnHelp$ = "IData"
            CASE CHR$(51),CHR$(27)
              Help$ = "Q"
          END SELECT
          IF INSTR(1,"CQS",Help$) <> 0 THEN EXIT LOOP
        LOOP
      LOOP UNTIL Help$ <> "C"
      IF Help$ <> "Q" THEN
        'store the current highway section in the dato$ file under SetNum&
1930    SetNum& = INT((LOF(1)/197) + 1)
        PUT # 1, SetNum&
      END IF
      CLOSE
      ERASE dc$,Identification$,ColArr%
RETURN
'---------------------------------------------------------------------------
'the following routine is used to input the values for structural number (SN),
'roadbed resilient modulus (Mr), reliability (R), and standard deviation (So)
'first, create the input screen
AASHTO:
     Editing$ = "A" : SN$ = "" : Reliability$ = "" : OnHelp$ = "AASHTO"
     CALL StrDatScr
'input the resilient modulus
     DO
       CALL MenuChoice(Choice$,9,49)
       choice% = VAL(Choice$)
1955   Help$ = ""
       SELECT CASE Choice$
         CASE CHR$(49)                ' enter the Mr value from the keyboard
           DO
1965         LOCATE 10,48,0  :  PRINT STRING$(10,0)
             CALL InputDouble(Value#,10,49,8)
             IF INSTR(1,"CQ",Help$) <> 0 THEN EXIT LOOP
             ResilientModulus# = Value#
           LOOP UNTIL ResilientModulus# <> 0
           EXIT LOOP
```

```
        CASE CHR$(50)
          DO
1970        LOCATE 11,48,0  :  PRINT STRING$(5,0)
            CALL InputDouble(Value#,11,49,3)
            IF INSTR(1,"CQ",Help$) <> 0 THEN EXIT LOOP
            CBRvalue# = Value#
          LOOP UNTIL CBRvalue# <> 0
          IF INSTR(1,"CQ",Help$) = 0 THEN
            'the correlation used is equation 1.5.1 of the 1986 AASHTO Guide
            'for the Design of Pavement Structures : Mr = 1500 * CBR    (psi)
            ResilientModulus# = 1500 * CBRvalue#
            LOCATE 11,57,0
            PRINT USING " Mr = ########"; ResilientModulus#
          END IF
          EXIT LOOP
        CASE CHR$(51)
          DO
1975        LOCATE 12,48,0  :  PRINT STRING$(5,0)
            RowDat% = 12  :  ColDat% = 49
            CALL InputDouble(Value#,12,49,3)
            IF INSTR(1,"CQ",Help$) <> 0 THEN EXIT LOOP
            Rvalue# = Value#
          LOOP UNTIL Rvalue# <> 0
          IF INSTR(1,"CQ",Help$) = 0 THEN
            'the correlation used is equation 1.5.3 of the 1986 AASHTO Guide
            'for the Design of Pavement Structures :  Mr = 1000 + (555)(R-value)
            ResilientModulus# = 1000 + (555 * Rvalue#)
            LOCATE 12,57,0
            PRINT USING " Mr = ########"; ResilientModulus#
          END IF
          EXIT LOOP
        CASE CHR$(27)
          ERASE ESALu!
          Help$ = "Q"
      END SELECT
    LOOP UNTIL Help$ = "Q"
    IF Help$ <> "Q" THEN
      DO
        DO
2010      LOCATE 14,48,0  :  PRINT STRING$(7,0)
          CALL InputReal(Value!,14,49,5)
        LOOP UNTIL Help$ <> "C"
        IF Help$ <> "Q" THEN
          Relity! = Value!
          Reliability$ = "Y"
          SELECT CASE Relity!
            CASE 0, 50 TO 99.99
              IF Relity! = 0 THEN       ' input the default value
                Relity! = 95            ' a default of 95 % is assumed
                LOCATE 14,49,0
                PRINT Relity!
              END IF
```

```
              CALL Reliability(Relity!,StandDev!)
              EXIT LOOP
          END SELECT
        END IF
      LOOP UNTIL Help$ = "Q"
      IF Help$ <> "Q" THEN
2100    LOCATE 14,57,0  :  PRINT USING " Zr =  +#.### "; StandDev!
        DO
2105      LOCATE 17,48,0  :  PRINT STRING$(7,0)
          CALL InputReal(Value!,17,49,5)
        LOOP UNTIL Help$ <> "C"
        IF Help$ <> "Q" THEN
          StandardDeviation! = Value!
          'input the default value of 0.45 for standard deviation
          IF StandardDeviation! = 0 THEN StandardDeviation! = 0.45
          'erase the relarray array before leaving
          ERASE RelArr!
          'store the results in the dato$ file
          LSET modulus$ = MKD$(ResilientModulus#)
          LSET deviate$ = MKS$(StandDev!)
          LSET deviation$ = MKS$(StandardDeviation!)
          'input the data for the computation of structural number, refer to
          'section 2.3.5 of the 1986 AASHTO Guide for the Design of Pavement
          'Structures
          SN$ = "Y"     ' create the data input screen
          CALL StrDatScr
          DO
            CALL MenuChoice(Choice$,13,38)
            SNchoice% = VAL(Choice$)
            DO
              SELECT CASE Choice$
                CASE CHR$(49)
                  CALL SNcoeff
                  'if the choice was changed during execution of the
                  'StrDatEd routine, then go back and execute the new choice
                  ERASE Layer$,LayerCoef!
                  IF SNchoice% <> 2 THEN EXIT LOOP ELSE Choice$ = CHR$(50)
                CASE CHR$(50)
                  CALL SNdesign
                  IF SNchoice% <> 1 THEN EXIT LOOP ELSE Choice$ = CHR$(49)
                CASE CHR$(27)
                  Help$ = "Q"
              END SELECT
            LOOP UNTIL Help$ = "Q"
            IF Help$ = "Q" THEN ERASE ESALu!
            EXIT LOOP
          LOOP
        ELSE
          ERASE ESALu!
        END IF
        IF Help$ <> "Q" THEN
          'determine what type of performance measure to use
```

```
            Dscr$ = "N"
 2250       CALL PerfScr   ' for selecting a Performance Measure
 2255       COLOR ColArr%(12),ColArr%(15)
            DO
              Help$ = "N"
              CALL MenuChoice(Choice$,8,38)
              PerfChoice% = VAL(Choice$)
              SELECT CASE Choice$
                CASE CHR$(49)
                  'check that deflection was not selected
                  IF OnDefl$ = "N" THEN
                    CALL PefWait("Y")
                    CALL PSI ' compute the values for PSI
                    PSIindex$ = "Y"         ' compute the PSI-ESAL index and
                    CALL PBAindex
                    'clean the memory to be saved
                    DIM DYNAMIC Deflection!(17:22)
                    deflection! = 0
                    CALL SaveDfl
                    ERASE Deflection!
                    Help$ = "S"
                    OnDefl$ = ""
                  END IF
                CASE CHR$(50)
                  CALL DeflectionData
                  IF Help$ <> "Q" THEN
                    CALL PefWait("Y")
                    CALL DeflectionIndex
                    ERASE Deflection!
                    Help$ = "S"
                  END IF
                CASE CHR$(51),CHR$(27)
                  ERASE Xval!,Yval!
                  Help$ = "C"
              END SELECT
            LOOP UNTIL INSTR(1,"SC",Help$) <> 0
          END IF
        ELSE
          ERASE ESALu!
        END IF
      END IF
      RETURN   'return to line 1930
'..........................................................................
'input the reliability value, create un array for reliabilities between
'90% and 99%
SUB Reliability(Relity!,StandDev!)
2000  DIM DYNAMIC RelArr!(90:99)
        RelArr!(90) = -1.282
        RelArr!(91) = -1.340
        RelArr!(92) = -1.405
        RelArr!(93) = -1.476
        RelArr!(94) = -1.555
```

```
        RelArr!(95) = -1.645
        RelArr!(96) = -1.751
        RelArr!(97) = -1.881
        RelArr!(98) = -2.054
        RelArr!(99) = -2.327
'determine the value of the standard normal deviate based on Table 4.1 of
'the 1986 AASHTO Guide for the Design of Pavement Structures
      IF Relity! >= 50 AND Relity! < 60 THEN
        StandDev! = (-0.0253 * (Relity! - 50)) - 0.000
      ELSEIF Relity! >= 60 AND Relity! < 70 THEN
        StandDev! = (-0.0271 * (Relity! - 60)) - 0.253
      ELSEIF Relity! >= 70 AND Relity! < 75 THEN
        StandDev! = (-0.0300 * (Relity! - 70)) - 0.524
      ELSEIF Relity! >= 75 AND Relity! < 80 THEN
        StandDev! = (-0.0334 * (Relity! - 75)) - 0.674
      ELSEIF Relity! >= 80 AND Relity! < 85 THEN
        StandDev! = (-0.0392 * (Relity! - 80)) - 0.841
      ELSEIF Relity! >= 85 AND Relity! < 90 THEN
        StandDev! = (-0.0490 * (Relity! - 85)) - 1.037
      ELSEIF Relity! >= 90 AND Relity! <= 99 THEN
        StandDev! = RelArr!(Relity!)
      ELSEIF Relity! > 99.0 AND Relity! <= 99.9 THEN
        StandDev! = (-0.8478 * (Relity! - 99.0)) - 2.327
      ELSEIF Relity! > 99.9 AND Relity! <= 99.99 THEN
        StandDev! = (-7.3333 * (Relity! - 99.9)) - 3.090
      END IF
      ERASE RelArr!
END SUB
'.........................................................................
'the following subroutine enters the data for structural Layer coefficients
'(ai), Layer thickness (Di), and drainage coefficients (mi), and then
'computes the value for structural number
SUB SNcoeff
      SHARED ColArr%(),LayerCoef!(),Layer%,LayerIN%,Help$,SNchoice%
      SHARED StructuralNumber!,StructuralNumber$,OnHelp$
      OnHelp$ = "AASHTO"
      COLOR ColArr%(12),ColArr%(15)        ' create the input screen
      LOCATE 16,30,0  :  PRINT "    ai     "
      LOCATE 16,41,0  :  PRINT "    Di     "
      LOCATE 16,52,0  :  PRINT "    mi     "
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 17,14,0  :  PRINT "Surface Course"
      LOCATE 18,14,0  :  PRINT "Base Layer"
      LOCATE 19,14,0  :  PRINT "Subbase Layer"
      DIM DYNAMIC Layer$(17:19),LayerCoef!(17:19,17:19)
        Layer$(17) = " Surface Course "
        Layer$(18) = " Base Layer     "
        Layer$(19) = " Subbase Layer  "
'the next loop is used to move the cursor between columns, and enter the data
      FOR Layer% = 17 TO 19
        COLOR ColArr%(12),ColArr%(15)
        LOCATE Layer%,13,0  :  PRINT Layer$(Layer%)
```

```
        NxtCol% = 30
        FOR LayerIN% = 17 TO 19
          IF Layer% = 17 AND LayerIN% = 19 THEN
            EXIT IF
          ELSE
2225        LOCATE Layer%,NxtCol%,0  :  PRINT STRING$(10,0)
            RowDat% = Layer%  :  ColDat% = NxtCol% + 1
            CALL InputReal(Value!,RowDat%,ColDat%,6)
            IF Help$ = "Q" THEN EXIT FOR
            LayerCoef!(Layer%,LayerIN%) = Value!
            IF SNchoice% = 2 THEN EXIT FOR
            LOCATE Layer%,NxtCol%,0
            PRINT USING " ##.###   "; LayerCoef!(Layer%,LayerIN%)
2230        INCR NxtCol%,11
          END IF
        NEXT LayerIN%
        IF Help$ = "Q" THEN EXIT FOR
        IF SNchoice% = 2 THEN EXIT FOR
        COLOR ColArr%(15),ColArr%(1)
        LOCATE Layer%,13,0  :  PRINT Layer$(Layer%)
      NEXT Layer%     ' move to the following Layer
'compute the structural number
      IF Help$ <> "Q" THEN
        IF SNchoice% <> 2 THEN
          StructuralNumber! = (LayerCoef!(17,17) * LayerCoef!(17,18))_
                            + (LayerCoef!(18,17) * LayerCoef!(18,18)_
                            * LayerCoef!(18,19)) + (LayerCoef!(19,17)_
                            * LayerCoef!(19,18) * LayerCoef!(19,19))
          LSET StructuralNumber$ = MKS$(StructuralNumber!)
        END IF
2235  END IF
END SUB
'..........................................................................
' the following subroutine asks the user to enter a value for SN
SUB SNdesign
      SHARED ColArr%(),Help$,StructuralNumber!,SNchoice%,StructuralNumber$
      SHARED OnHelp$
      OnHelp$ = "AASHTO"
      COLOR ColArr%(12),ColArr%(15)
2240  LOCATE 17,12,0 : PRINT " Enter a value for Structural Number ==> "
2245  Help$ = ""
      DO
        LOCATE 17,55,0 : PRINT " SN =        "
        CALL InputReal(Value!,17,62,4)
        IF HELP$ = "Q" THEN EXIT LOOP
        StructuralNumber! = Value!
        IF SNchoice% = 1 THEN EXIT LOOP
      LOOP UNTIL StructuralNumber! <> 0
      IF Help$ <> "Q" THEN
        IF SNchoice% <> 1 THEN LSET StructuralNumber$ = MKS$(StructuralNumber!)
2246  END IF
END SUB
```

```
'-----------------------------------------------------------------------------
'this subroutine computes delta PSI (change in Present Serviceability Index)
'based on the data input through and/or computed by the former subroutines,
'this routine is based on equation 1.2.1, page I-5 of the 1986 AASHTO Guide
'for the Design of Pavement Structures
SUB PSI
       SHARED PSI!(),StandDev!,StandardDeviation!,StructuralNumber!
       SHARED ResilientModulus#,NoData%,ESALu!(),PSI1!,PSI2!,PSI3!
       DIM DYNAMIC PSI!(0:10)
       PSI1! = ((StandDev!)*(StandardDeviation!))
       PSI2! = (9.36 * (LOG10(StructuralNumber! + 1))) - 0.20
5010   PSI3! = (2.32 * (LOG10(ResilientModulus#))) - 8.07
       PSI4! = PSI1! + PSI2! + PSI3!
       PSI5! = (0.40 + (1094/((StructuralNumber! + 1)^5.19)))
       PSIcounter% = 1
       PSIstop% = NoData%
       PSIarray% = 1
'the following loop computes the PSI value for each value of ESAL
       DO UNTIL PSIcounter% > PSIstop%
         W18# = ESALu!(PSIcounter%)
'for ESAL = 0 skip the error obtained from LOG10(0)
         IF W18# = 0 THEN
           PSI6! = 4.2
         ELSE
5020       deltaPSI! = (EXP10(((LOG10(W18#)) - PSI4!) * PSI5!)) * (2.7)
           PSI6! = 4.2 - deltaPSI!
         END IF
5030     PSI!(PSIarray%) = PSI6!
         INCR PSIcounter%
         INCR PSIarray%
       LOOP
'store the PSI values in the dato$ file
       PSI!(0) = 4.2
END SUB
'-----------------------------------------------------------------------------
'the following subroutine is used to enter the data needed to compute the
'deflection on the pavement structure.  Refer to Appendix PP of the 1986
'AASHTO Guide for the Design of Pavement Structures.
SUB DeflectionData
12400 SHARED Editing$,Deflection!(),ColArr%(),Help$,DynamicLoad$,PlateRadius$
      SHARED PavementThickness$,SubgradePoisson$,FieldSN$,DeflectionFactor$
      SHARED DeflSN$,DeflDesign$,d9!,StructuralNumber!,deflection!,OnHelp$
      'display the input screen
      Editing$ = "D" : OnHelp$ = "DEFLECTION"
      CALL DefScr
      'save the instructions in an array
      DIM DYNAMIC Deflection$(17:22), Deflection!(17:22)
      Deflection$(17) = " Dynamic Load < P, lbs> : "
      Deflection$(18) = " Load Plate Radius < ac, inch> : "
      Deflection$(19) = " Pavement Thickness < h, inch> : "
      Deflection$(20) = " Subgrade's Poisson Ratio < u > : "
      Deflection$(21) = " Field Structural Number : "
```

```
      Deflection$(22) = " Design Deflection Value : "
      COLOR ColArr%(15),ColArr%(1)
      'display the data needed
      FOR D% = 17 TO 22
        LOCATE D%,15,0
        PRINT Deflection$(D%)
      NEXT D%
      'enter the data needed
      FOR D% = 17 TO 22
        COLOR ColArr%(12),ColArr%(15)
        LOCATE D%,15,0  :  PRINT Deflection$(D%)
        LOCATE D%,50,0  :  PRINT SPACE$(10)
        CALL InputReal(Value!,D%,52,8)
        IF Help$ = "Q" THEN EXIT FOR
        Deflection!(D%) = Value!
        COLOR ColArr%(15),ColArr%(1)
        LOCATE D%,15,0
        PRINT Deflection$(D%)
      NEXT D%
      IF Help$ <> "Q" THEN
        SN! = StructuralNumber!
        deflection! = FNDeflection!(SN!)
        CALL SaveDfl
      END IF
      ERASE Deflection$
END SUB
'..........................................................................
'save the defelction values
SUB SaveDfl
12425 SHARED Deflection!(),DynamicLoad$,PlateRadius$,PavementThickness$
      SHARED SubgradePoisson$,FieldSN$,DeflectionFactor$,DeflSN$,DeflDesign$
      SHARED d9!,deflection!
      LSET DynamicLoad$ = MKS$(Deflection!(17))
      LSET PlateRadius$ = MKS$(Deflection!(18))
      LSET PavementThickness$ = MKS$(Deflection!(19))
      LSET SubgradePoisson$ = MKS$(Deflection!(20))
      LSET FieldSN$ = MKS$(Deflection!(21))
      LSET DeflectionFactor$ = MKS$(d9!)
      LSET DeflSN$ = MKS$(deflection!)
      LSET DeflDesign$ = MKS$(Deflection!(22))
END SUB
'--------------------------------------------------------------------------
'the following subroutine plots in the screen performance vrs independent
'variable, and then computes the area between the initial performance value and
'the curve, this area is identified as the PBA Index.
SUB PBAindex
      SHARED MaxESAL!,Xmin!,ESALu!(),XvalMax%,MaxW18%,Xval!(),Yval!(),ColArr%()
      SHARED AreaBetween!,PBAindex$,HighwayNo$,PerformanceM$,UnitsX$,Xpoint!
      SHARED Xdiv%,Xdiv1%,XlagI%,XlagF%,X0%,ExpX%,PSI!(),Ymax!,Ymin!,PfDesign!
      SHARED NoData%,Help$,User$,PlotDef$,Identification$(),OnHelp$,Xm!
      OnHelp$ = "PBAindex"
      MaxESAL! = 0 : Xmin! = 0
```

```
9495  FOR MaxW18% = 1 TO 10
        IF MaxESAL! <= ESALu!(MaxW18%) THEN
          MaxESAL! = ESALu!(MaxW18%)
          XvalMax% = MaxW18%
        END IF
        IF Xmin! >= ESALu!(MaxW18%) THEN Xmin! = ESALu!(MaxW18%)
        Xval!(MaxW18%) = ESALu!(MaxW18%)      'these are the values to be
        Yval!(MaxW18%) = PSI!(MaxW18%)        'plotted by PlotGraph
      NEXT MaxW18%
'compute the area between the curve and a design performance value
      IF User$ = "N" THEN CALL IndexArea(AreaBetween!)
      IF AreaBetween! <> 0 THEN
        LSET PBAindex$ = MKS$(AreaBetween!)
        LOCATE 11,30,0
        PRINT USING "Highway ID = & "; Identification$(8); : PRINT " "
        LOCATE 12,11,0  :  PRINT SPACE$(55)
        LOCATE 12,10,0
        PRINT UCASE$(PerformanceM$) + CHR$(45) + MID$(UnitsX$,1,4) + " Index = ";
        COLOR ColArr%(12),ColArr%(15)
        'print the performance-? index
        IF AreaBetween! < 1 THEN
          PRINT USING " ##.######^^^^ "; AreaBetween!;
        ELSE
          PRINT USING " ########## "; AreaBetween!;
        END IF
        COLOR ColArr%(15),ColArr%(1)
        PRINT " " + UCASE$(PerformanceM$) + " *  " + MID$(UnitsX$,1,4)
        LOCATE 13,10,0
        PRINT "Do you want to plot the " + UCASE$(PerformanceM$);
        PRINT " vs " + MID$(UnitsX$,1,4) + " curve (y/n)? : "
        DO
          CALL MenuChoice(Choice$,13,64)
          COLOR ColArr%(15),ColArr%(1)
          SELECT CASE UCASE$(Choice$)
            CASE "Y"
              PlotDef$ = ""
              'determine the X scale
              CALL Xscale(MaxESAL!,ExpX%,Xval!(),Xpoint!)
              XlagI% = 0 : XlagF% = 10 : Xdiv% = 50 : Xdiv1% = 1 : X0% = 0
              Xm! = Xmin!
              CALL PlotGraph
              EXIT LOOP
            CASE "N"
              EXIT LOOP
          END SELECT
        LOOP
        SCREEN 0
      END IF
      ERASE ESALu!,PSI!
END SUB
'.............................................................................
'define the X scale
```

```
SUB Xscale(MaxESAL!,ExpX%,Xval!(1),Xpoint!)
      'determine the X scale
      IF MaxESAL! < 1 THEN
        Flow% = -1 : Bound% = -10
      ELSE
        Flow% = 1  : Bound% = 10
      END IF
      FOR ExpX% = 0 TO Bound% STEP Flow%
        IF MaxESAL! > 1 THEN
          IF (MaxESAL! / (10^(ExpX%))) < 1 THEN
            DECR ExpX%
            EXIT FOR
          END IF
        ELSE
          IF (MaxESAL! / (10^(ExpX%))) > 1 THEN
            'transform the values of X for graphics purposes
            FOR Tr% = 1 TO 10
              Xval!(Tr%) = Xval!(Tr%) / (10^(ExpX%))
            NEXT Tr%
            MaxESAL! = MaxESAL! / (10^(ExpX%))
            ExpX% = 0
            EXIT FOR
          END IF
        END IF
      NEXT ExpX%
      Xpoint! = MaxESAL! / (10^(ExpX%))
END SUB
'.......................................................................
'the following subroutine plots in the screen Structural Number vrs Deflection
'and then computes the area between the initial PSI and the curve, this area
'is identified as the Deflection Index
SUB DeflectionIndex
37000 SHARED Xmin!,XvalMax%,Xval!(),Yval!(),AreaBetween!,PBAindex$,HighwayNo$
      SHARED PerformanceM$,UnitsX$,ColArr%(),Xpoint!,Xdiv%,Xdiv1%,XlagI%,XlagF%
      SHARED X0%,Ymax!,Ymin!,PfDesign!,NoData%,Help$,User$,PlotDef$,LastSN%
      SHARED Identification$(),StructuralNumber!,PSIindex$,Deflection!()
      SHARED OnHelp$,ExpX%,Xm!,MaxESAL!
      'compute the area between the curve and a design value
      PSIindex$ = "N"   : OnHelp$ = "DefIndex"
      IF Deflection!(21) >= StructuralNumber! THEN
        LOCATE 12,20,0  :  PRINT SPACE$(50)
        LOCATE 12,27,0
        PRINT "Error on deflection data "
        LOCATE 13,19,0
        PRINT "Press ESC to enter data for a new section"
        DO
          NoArea$ = INKEY$
        LOOP UNTIL NoArea$ = CHR$(27)
        Help$ = "Q"
37100 ELSE
        CALL IndexArea(AreaBetween!)
        IF AreaBetween! <> 0 THEN
```

```
          LSET PBAindex$ = MKS$(AreaBetween!)
          LOCATE 11,30,0
          PRINT USING "Highway ID: & "; Identification$(8); : PRINT " "
          LOCATE 12,11,0  :  PRINT SPACE$(55)
          LOCATE 12,10,0  :  PRINT UCASE$(PerformanceM$) + " SN - Index =  "
          LOCATE 12,30,0  :  COLOR ColArr%(12),ColArr%(15)
          PRINT USING "###.####### "; AreaBetween!;
          COLOR ColArr%(15),ColArr%(1)
          PRINT " inch * SN "
          LOCATE 13,10,0
          PRINT "Do you want to plot the " + UCASE$(PerformanceM$) +_
                                      " vs SN curve (y/n)? : "
37200     DO
            CALL MenuChoice(Choice$,13,64)
            plot$ = UCASE$(Choice$)
            SELECT CASE plot$
              CASE "Y"
                FirstSN% = INT(StructuralNumber!)
                Xdiv% = 5  :  Xdiv1% = 1
                FieldSN! = Deflection!(21)
                FieldDef! = FNDeflection!(FieldSN!)
                LastSN% = INT(FieldSN!)
                FOR D! = 0 TO 1 STEP 0.01  ' determine the scale in the Y axis
                  Daxis! = D! + 0.01
                  IF D! <= FieldDef! AND FieldDef! < Daxis! THEN EXIT FOR
                NEXT D!
                Ymax! = Daxis! : Ymin! = 0
                PlotDef$ = "Y"
                PerformanceM$ = "DEFLECTION"
                UnitsX$ = "SN"
                PfDesign! = Deflection!(22)
                Xmin! = 0
                Xm! = Deflection!(21)
                NoData% = 1
                Xval!(1) = Deflection!(21)
                Yval!(1) = FNDeflection!(Deflection!(21))
                SN! = LastSN%
37300           DO
                  INCR SN!
                  INCR NoData%
                  Xval!(NoData%) = SN!
                  Yval!(NoData%) = FNDeflection!(SN!)
                LOOP UNTIL SN! >= StructuralNumber!
                Xval!(NoData%) = StructuralNumber!
                Yval!(NoData%) = FNDeflection!(StructuralNumber!)
                XvalMax% = NoData%  :  XlagI% = 0
                'determine the X scale
                MaxESAL! = StructuralNumber!
                CALL Xscale(MaxESAL!,ExpX%,Xval!(),Xpoint!)
                XlagI% = 0 : XlagF% = 10 : Xdiv% = 50 : Xdiv1% = 1 : X0% = 0
                CALL PlotGraph
                EXIT LOOP
```

```
              CASE "N"
                EXIT LOOP
37400         END SELECT
              IF NoArea$ = CHR$(27) THEN EXIT LOOP
            LOOP
          END IF
          SCREEN 0
        END IF
END SUB
'.........................................................................
'the following subroutine computes the area between the line representing a
'constant design value and the Performance curve, the Romberg method of
'numerical integration is used for this purpose
SUB IndexArea(AreaBetween!)
      SHARED PSIindex$,MaxESAL!,StructuralNumber!,Deflection!()
'define the two extreme values on the X axis
9610  IF PSIindex$ = "Y" THEN     ' using PSI data
        RiBou! = MaxESAL!
        LeftBoundary! = 0
      ELSE   ' using deflection data
        RiBou! = StructuralNumber!
        LeftBoundary! = Deflection!(21)
      END IF
'call the subroutine used to compute the area
9611  CALL Integration(LeftBoundary!,RiBou!,AreaBeneath!,PSIindex$)
      IF AreaBeneath! <> 0 THEN
        IF PSIindex$ = "Y" THEN
'determine the area between the axis and a constant line at PSI = 4.2
9612      TotalArea! = 4.2 * MaxESAL!
'compute the area between the line at PSI= 4.2 and the PSI-ESAL curve
          AreaBetween! = TotalArea! - AreaBeneath!
          PSIindex$ = ""
        ELSE
'determine the area between the axis and a constant line at the deflection
'for the design Structural Number
9613      ConstantArea! = (StructuralNumber!-Deflection!(21)) * Deflection!(22)
'compute the area between the line  and the deflection - SN curve
9614      AreaBetween! = AreaBeneath! - ConstantArea!
        END IF
      ELSE
        AreaBetween! = 0
      END IF
END SUB
'.........................................................................
'the following subroutine is the integration by the Romberg method
SUB Integration(LeftBoundary!,RiBou!,AreaBeneath!,PSIindex$)
'if the user entered ESAL = 0
    IF RiBou! = 0 THEN
      AreaBeneath! = 0
      EXIT SUB
    END IF
'input a tolerance value to stop the iteration procedure
```

```
    tolerance = 1.0E-5
'the following arrays store the number of divisions used on each iteration
'(Counter!), and the areas and interpolated values used (Area!)
    DIM DYNAMIC Counter!(128), Area!(256)
'start with 1 trapezoid
    nDivisions! = 1
    Counter!(1) = 1
    deltaX = (RiBou! - LeftBoundary!) / nDivisions!
    IF PSIindex$ = "Y" THEN
      BorderSum! = (4.2 + FNpsi!(RiBou!)) / 2
      Area!(1) = deltaX * BorderSum!
    ELSE
      BorderSum! = (FNDeflection!(LeftBoundary!) +_
                              FNDeflection!(RiBou!))/2
    END IF
    Interpol% = 1
    InterpolNext% = 2
    IndexSum! = BorderSum!
'for more than 1 trapezoid
      DO
        INCR Interpol%
        power4 = 4
        Counter!(Interpol%) = InterpolNext%
        nDivisions! = nDivisions! * 2
        Odd = nDivisions! - 1
        deltaX = (RiBou! - LeftBoundary!) / nDivisions!
'the next loop determines the sum of the PSI or Deflection values for
'ESAL values or Structural Number values respectively
        FOR OddNext = 1 TO (Odd + 1)/2
          I = (OddNext * 2) - 1
          X = LeftBoundary! + (I * deltaX)
          IF PSIindex$ = "Y" THEN
            IndexSum! = IndexSum! + FNpsi!(X)
          ELSE
            IndexSum! = IndexSum! + FNDeflection!(X)
          END IF
        NEXT OddNext
'compute the area given nDivisions!
        Area!(InterpolNext%) = IndexSum! * deltaX
        Test1 = Counter!(Interpol% - 1)
        Z = Interpol% - 1
'the following FOR block computes the interpolated values
        FOR B = 1 TO Z
          Jbelow = InterpolNext% + B
          Jabove = Counter!(Interpol% - 1) + B - 1
          Area!(Jbelow) = ((power4 * Area!(Jbelow - 1)) -_
                               Area!(Jabove)) /(power4 - 1)
          power4 = power4 * 4
        NEXT B
'compare differences with the accepted tolerance
        IF (Interpol% > 4) AND (Area!(InterpolNext% + 1) <> 0) THEN
          Check1 = ABS(Area!(Test1) - Area!(InterpolNext% + 1))
```

```
          Check2 = ABS(Area!(InterpolNext% + 1) * tolerance)
'for the first interpolated area values of two consecutive iterations
          IF Check1 <= Check2 THEN EXIT LOOP
          Check3 = ABS(Area!(InterpolNext% - 1) - Area!(Jbelow))
          Check4 = ABS(Area!(Jbelow) * tolerance)
'for the last interpolated area values of two consecutive iterations
          IF Check3 <= Check4 THEN EXIT LOOP
          IF (Interpol% > 16) THEN
            LOCATE 12,19,0
            PRINT " No convergence was reached with these data "
            LOCATE 13,19,0
            PRINT " Press ESC to enter data for a new section  "
            Area!(Jbelow) = 0
            DO
              NoArea$ = INKEY$
            LOOP UNTIL NoArea$ = CHR$(27)
          END IF
        END IF
        IF NoArea$ = CHR$(27) THEN EXIT LOOP
        InterpolNext% = Jbelow + 1
      LOOP  ' end the determination of interpolated values
9640  AreaBeneath! = Area!(Jbelow)
      ERASE Counter!,Area!
END SUB    ' return to the PBAindex routine
'..........................................................................
'define the function to be integrated, equation 1.2.1 of the 1986 AASHTO Guide
'for the Design of Pavement Structures is solved for delta-PSI and then the
'result substracted from an initial PSI value of 4.2
DEF FNpsi!(ESAL!)
      SHARED StandDev!,StandardDeviation!,StructuralNumber!,ResilientModulus#
      SHARED PSI1!,PSI2!,PSI3!
      LOCAL psiesal1!, psiesal2!, psiesal3!, psiesal4!, psiesal5!, psi!
      psiesal1! = ((StandDev!)*(StandardDeviation!))
      psiesal2! = (9.36 * (LOG10(StructuralNumber! + 1))) - 0.20
9620  psiesal3! = (2.32 * (LOG10(ResilientModulus#))) - 8.07
      psiesal4! = PSI1! + PSI2! + PSI3!
      psiesal5! = (0.40 + (1094/((StructuralNumber! + 1)^5.19)))
9625  psi! = 4.2 - ((EXP10(((LOG10(ESAL!)) - psiesal4!) * psiesal5!)) * (2.7))
9630  FNpsi! = psi!
END DEF
'..........................................................................
'the following function computes the theoretical deflection, refer to
'equation PP.26, Appendix PP, of the 1986 AASHTO Guide for the Design of
'Pavement Structures
DEF FNDeflection!(SN!)
15000 SHARED Deflection!(),ResilientModulus#,d9!
      d1! = 1 - ((Deflection!(20))^2)
      d2! = (d1! / (ResilientModulus#))^(1/3)
      d3! = (209.3 * SN! * d2!) / Deflection!(18)
      d4! = (1 + ((d3!)^2))^0.5
      d5! = 1 - Deflection!(20)
      d6! = 2 * d5! * d4!
```

```
        d7! = 1 + (d3! / d6!)
        d8! = ((1 + ((d3!)^2))^0.5) - d3!
        d9! = d8! * d7!
        d10! = (0.0043 * Deflection!(19))^3
        d11! = ResilientModulus# * d10!
        d12! = (((SN!)^3 * d1!) / d11!) - 1
        d13! = 1 + (d9! * d12!)
        d14! = 2 * Deflection!(17) * d10!
        d15! = 3.1416 * Deflection!(18) * ((SN!)^3)
        d16! = (d14! / d15!) * d13!
        FNDeflection! = d16!
END DEF
'-----------------------------------------------------------------------------
'the following subroutine plots the performance vrs independent variable curve
SUB PlotGraph
35000 SHARED Xmin!,ESALu!(),XvalMax%,MaxW18%,MaxESAL!,Xval!(),Yval!()
      SHARED AreaBetween!,PBAindex$,HighwayNo$,PerformanceM$,UnitsX$,ColArr%()
      SHARED Xpoint!,Xdiv%,Xdiv1%,XlagI%,XlagF%,X0%,ExpX%,PSI!(),Ymax!,Ymin!
      SHARED VideoGraph%,PfDesign!,NoData%,Help$,User$,PlotDef$,OnGraph$
      SHARED Identification$(),LastSN%,StructuralNumber!,VideoMode%,Xm!
      'retrieve the graphics dbase
      'PlGu%( ,1)=Xview%; PlGu%( ,2)=Xscale%; PlGu%( ,3)=Xmax!; PlGu%( ,4)=Dot!
      'PlGu%( ,5)=Xscale% for Hercules
      DIM DYNAMIC PlGu%(1:10,1:5)
      'read the database for graphics
      RESTORE 9900
      FOR Rrow% = 1 TO 10
        FOR Rcol% = 1 TO 5
          READ PlGu%(Rrow%,Rcol%)
        NEXT Rcol%
      NEXT Rrow%

35100 FOR Xlag% = XlagI% TO XlagF%
        'determine the number of divisions in the X-axis
        IF Xpoint! >= 0 AND Xpoint! <= Xlag% THEN
          'determine the number of pixels between each X-axis division
          IF VideoMode% = 7 THEN
            Xscale% = PlGu%(Xlag%,5)
            Xspace% = Xscale%/9
            Xview% = (PlGu%(Xlag%,5) * Xlag%) + 212
          ELSE
            Xscale% = PlGu%(Xlag%,2)
            Xspace% = Xscale%/8
            Xview% = PlGu%(Xlag%,1)
          END IF
          Xmax! =  PlGu%(Xlag%,3) * (10^(ExpX%-1))
          EXIT FOR
        END IF
      NEXT Xlag%
35200 'determine if there is any error on graphics data
      Help$ = "Q"
      IF Xmax! >= Xval!(XvalMax%) AND Xmax! > Xmin! THEN
```

```
        IF Ymax! > Ymin! THEN
          IF Xmin! >= 0 AND Ymin! >= 0 THEN
            OnGraph$ = "Y"  'flag for error routine
            SELECT CASE VideoGraph%
              CASE 16
                SCREEN 9   'EGA
              CASE 14
                SCREEN 8   'CGA color
              CASE 6,7
                SCREEN 2   'Hercules, CGA monochrome
              CASE ELSE
                EXIT IF
            END SELECT
            IF VideoGraph% = 16 OR VideoGraph% = 14 THEN
              COLOR ColArr%(15),ColArr%(1)
            END IF
            LOCATE 3,16,0
            PRINT UCASE$(PerformanceM$)+CHR$(45)+MID$(UnitsX$,1,4)+" Index = ";
            'print the performance-? index
            IF AreaBetween! < 1 THEN
              PRINT USING " ##.######^^^^ "; AreaBetween!;
            ELSE
              PRINT USING " ##########.#### "; AreaBetween!;
            END IF
            PRINT " " + UCASE$(PerformanceM$) + " *  " + MID$(UnitsX$,1,4)
            LOCATE 8,14,0  :  PRINT USING "###.#^^^^"; Ymax!
            LOCATE 18,14,0 :  PRINT USING "##.#^^^^"; Ymin!
            'print a label for y axis
            FOR I% = 1 TO 10
              I$ = MID$(PerformanceM$,I%,1)
              LOCATE (I%+7),13,0 : PRINT I$
            NEXT I%
35300       LOCATE 22,35,0
            PRINT USING "Highway ID = & "; Identification$(8) + " "
            LOCATE 24,30,0 : PRINT "Press any key to continue";
            LOCATE 20,53,0 : PRINT UCASE$(UnitsX$)
            'when using AASHTO PSI equation
            IF Ymax! = 5 THEN
              Yscale! = 4
              FOR Dash% = 10 TO 16 STEP 2
                LOCATE Dash%,21,0
                PRINT Yscale!
                Yscale! = Yscale! - 1
              NEXT Dash%
            END IF
            'print on the screen the values for each X-axis division
            ESALaxis% = X0%
            FOR Xdash% = 23 TO (23+(Xspace%*Xlag%)) STEP Xspace%
              LOCATE 19,Xdash%,0
              PRINT ESALaxis%
              ESALaxis% = ESALaxis% + Xdiv1%
            NEXT Xdash%
```

```
        'define the active graphics screen, and draw the axis
35400   SELECT CASE VideoGraph%
          CASE 16,14,6     'EGA, CGA color, CGA b&w
            IF VideoGraph% = 16 THEN RESTORE 9950 ELSE RESTORE 9975
            DIM DYNAMIC ECga%(1:12)
            FOR Rga% = 1 TO 12
              READ ECga%(Rga%)
            NEXT Rga%
            FOR C% = 188 TO Xview% STEP Xscale%
              LINE (C%,ECga%(1)) - (C%,ECga%(2)) ,ColArr%(15)
            NEXT C%
            LINE (40,ECga%(3)) - (600,ECga%(4)) ,ColArr%(15),B
            LINE (44,ECga%(5)) - (596,ECga%(6)) ,ColArr%(15),B
            LINE (88,ECga%(7)) - (550,ECga%(8)) ,ColArr%(12),B
            FOR L% = ECga%(9) TO ECga%(1) STEP ECga%(10)
              LINE (180,L%) - (188,L%) ,ColArr%(15)
            NEXT L%
            LINE (188,ECga%(11)) - (188,ECga%(9)) ,ColArr%(15)
            LINE (Xview%,ECga%(1)) - (Xview%+8,ECga%(1))
            VIEW (188,ECga%(1)) - (Xview%,ECga%(9)),ColArr%(1)
            LINE (0,0) - (0,ECga%(12)) ,ColArr%(15)
            LINE - ((Xview%-180),ECga%(12)) ,ColArr%(15)
            ERASE ECga%
          CASE 7       'HERCULES
            FOR C% = 212 TO (Xview%) STEP Xscale%
              LINE (C%,245) - (C%,252)
            NEXT C%
            LINE (45,8) - (675,344) ,,B
            LINE (49,9) - (671,342) ,,B
            LINE (99,70) - (619,315) ,,B
            FOR L% = 105 TO 245 STEP 28
              LINE (203,L%) - (212,L%)
            NEXT L%
            LINE (212,98) - (212,105)
            LINE (Xview%,245) - (Xview%+9,245)
            VIEW (212,245) - (Xview%,105)
            LINE (0,0) - (0,140)
            LINE - ((Xview%-194),140)
        END SELECT
        'the active screen will have world coordinates (Xmin!,Ymin!) as
        'origin, and (Xmax!,Ymax!) in the upper-right corner
35500   WINDOW (Xmin!,Ymin!) - (Xmax!,Ymax!)
        'plot the values
        SELECT CASE VideoGraph%
          CASE 14, 16
            'avoid graphics errors
            IF PfDesign! > 0 THEN
              LINE (Xm!,PfDesign!)-(Xval!(XvalMax%),PfDesign!),ColArr%(10)
            END IF
            PSET (Xval!(1),Yval!(1)) ,ColArr%(15)
            FOR T% = 2 TO NoData% STEP 1
              PSET (Xval!(T%),Yval!(T%)) ,ColArr%(15)
```

```
                LINE (Xval!(T%),Yval!(T%))-(Xval!(T%-1),Yval!(T%-1)),ColArr%(18)
              NEXT T%
            CASE ELSE
              IF PfDesign! > 0 THEN
                LINE (Xm!,PfDesign!) - (Xval!(XvalMax%),PfDesign!)
              END IF
              PSET (Xval!(1),Yval!(1))
              FOR T% = 2 TO NoData% STEP 1
                PSET (Xval!(T%),Yval!(T%))
                LINE (Xval!(T%),Yval!(T%)) - (Xval!(T%-1),Yval!(T%-1))
              NEXT T%
          END SELECT
          Help$ = "S"
        END IF
      END IF
    END IF
    'press any key to go back to the main data collection routine after
    'plotting the PSI-ESAL curve, or after an error in graphics data
35700 IF Help$ = "Q" THEN
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 11,10,0  :  PRINT SPACE$(60)
      LOCATE 12,10,0  :  PRINT SPACE$(60)
      LOCATE 13,10,0  :  PRINT SPACE$(60)
      LOCATE 13,27,0  :  PRINT "Press any key to continue"
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 12,16,0
      PRINT " Error in graphics input data or graphics card "
      DO : LOOP UNTIL INSTAT
      Null$ = INKEY$
    ELSE
      DO : LOOP UNTIL INSTAT
      Null$ = INKEY$
      VIEW
    END IF
    ERASE Xval!,Yval!,PlGu%
    OnGraph$ = ""
END SUB
'------------------------------------------------------------------------
SUB PefWait(Mesg$)
36000 SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 10,8,0 : PRINT CHR$(201) + STRING$(62,205) + CHR$(187)
      FOR I% = 11 TO 14
        LOCATE I%,8,0 : PRINT CHR$(186) + SPACE$(62) + CHR$(186)
      NEXT I%
      LOCATE 15,8,0 : PRINT CHR$(200) + STRING$(62,205) + CHR$(188)
      IF Mesg$ = "Y" THEN
        LOCATE 12,23,0 : PRINT "The PBA Index is being computed"
        COLOR ColArr%(16),ColArr%(10)
        LOCATE 13,37,0 :  PRINT " WAIT "
        COLOR ColArr%(15),ColArr%(1)
      END IF
```

```
END SUB
'--------------------------------------------------------------------------
'this routine shows the contents of a data file
ViewFile:
36100 CALL EqnScr(WorkFile$)
      RecNo& = LOF(1)/197
      FOR ViewDat& = 1 TO RecNo&
        GET # 1, ViewDat&
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 4,10,0 : PRINT USING "Record Number: ####"; ViewDat&
        LOCATE 5,10,0 : PRINT USING "Highway ID: &"; highwaynumber$
        LOCATE 6,10,0
        PRINT USING "PBA Index: +###########.######"; CVS(PBAindex$)
        LOCATE 8,5,0    : PRINT USING "District: &"; district$
        LOCATE 8,45,0   : PRINT USING "County: &"; county$
        LOCATE 9,5,0    : PRINT USING "Begin Mile Post: ###.###"; CVS(beginMP$)
        LOCATE 9,45,0   : PRINT USING "End Mile Post: ###.###"; CVS(endMP$)
        LOCATE 10,5,0   : PRINT USING "Section EA: &"; sectionEA$
        LOCATE 10,45,0 : PRINT USING "EA/SJ: &"; easj$
        LOCATE 11,5,0   : PRINT USING "Rehabilitation/Overlay Year: &"; year$
        LOCATE 11,45,0 : PRINT USING "Region: &"; region$
36200   LOCATE 13,5,0
        PRINT USING "Resilient Modulus: ########.##"; CVD(modulus$)
        LOCATE 13,45,0
        PRINT USING "Normal Deviate: +####.#####"; CVS(deviate$)
        LOCATE 14,5,0
        PRINT USING "Standard Deviation: ##.##"; CVS(deviation$)
        LOCATE 14,45,0
        PRINT USING "Structural Number: ##.##"; CVS(StructuralNumber$)
        LOCATE 17,5,0
        PRINT USING "Dynamic Load: #####.##"; CVS(DynamicLoad$)
        LOCATE 17,45,0
        PRINT USING "Plate Radius: ##.###"; CVS(PlateRadius$)
        LOCATE 18,5,0
        PRINT USING "Pavement Thickness: ##.##"; CVS(PavementThickness$)
        LOCATE 18,45,0
        PRINT USING "Subgrade Poisson Ratio: #.##"; CVS(SubgradePoisson$)
        LOCATE 19,5,0
        PRINT USING "Field Structural Number: ##.##"; CVS(FieldSN$)
        LOCATE 19,45,0
        PRINT USING "Deflection Factor: ###.#######"; CVS(DeflectionFactor$)
        LOCATE 20,5,0
        PRINT USING "As built Deflection: ###.######"; CVS(DeflSN$)
        LOCATE 20,45,0
        PRINT USING "Deflection Design: ##.######"; CVS(DeflDesign$)
36300   DO : LOOP UNTIL INSTAT
        MoreDat$ = INKEY$
        CALL ClScr(WorkFile$)
        IF MoreDat$ = CHR$(27) THEN EXIT FOR
      NEXT ViewDat&
RETURN
'--------------------------------------------------------------------------
```

```
'link the different routines which form PBA.EXE
      $INCLUDE "PBACAR.BAS"
      $INCLUDE "PBAEQN.BAS"
      $INCLUDE "PBAEDIT.BAS"
      $INCLUDE "PBAFILE.BAS"
      $INCLUDE "PBAHELP.BAS"
'------------------------------------------------------------------------
'database for graphics routine
9900  DATA 516, 328,  10,  6, 369
      DATA 508, 160,  20, 10, 180
      DATA 500, 104,  30, 15, 117
      DATA 508,  80,  40, 25,  90
      DATA 508,  64,  50, 30,  72
      DATA 524,  56,  60, 40,  63
      DATA 524,  48,  70, 50,  54
      DATA 508,  40,  80, 60,  45
      DATA 476,  32,  90, 60,  36
      DATA 508,  32, 100, 70,  36
'database for EGA
9950  DATA 245, 252, 14, 344, 16, 342, 70, 315, 105, 28, 98, 140
'database for CGA
9975  DATA 140, 144,  8, 196,  9, 195, 40, 180,  60, 16, 56,  80
'------------------------------------------------------------------------
'this is the end of the PBAASHTO.BAS program
CloseFile:
36400 CLOSE
      KILL "FILENAME.PBA"
      COLOR ColArr%(15),ColArr%(1)
      CLS
      END
'------------------------------------------------------------------------
'the following subroutine checks for a variety of errors
ErrorTrap:
      IF OnGraph$ = "Y" THEN SCREEN 0    'change to text screen
      SELECT CASE ERR
        CASE 5
          SELECT CASE ERL
            CASE 9780,9785  'checking the graphics card available
              SCREEN
              RESUME 9790
            CASE 9620,9625,9630  'logarithm of zero
              psi! = 4.2
              RESUME 9640
          END SELECT
        CASE 11    'division by zero
          SELECT CASE ERL
            CASE 3900
              RESUME 3910
          END SELECT
      END SELECT
      IF ERR <> 6 THEN DIM DYNAMIC ErMsg$(1:7)
      LOCATE 20,20,0   'turn cursor off
```

```
'variables used in the ErrowWindow routine
ErMsg$(4) = "Press ESC to continue" : ErMsg$(2) = "" : ErMsg$(5) = ""
ErMsg$(3) = "ERROR : System"  :  InHelp$ = "N"
SELECT CASE ERR
  CASE 6  'missing construction year
    CALL ErrorScreen
    NewYear$ = "Y"
    LOCATE 22,15,0
    PRINT " Enter the Construction/          "
    LOCATE 23,15,0
    PRINT " Rehabilitation Year:             "
    LOCATE 23,51,0
    PRINT "      "
    DO
      CALL InputString(StringIN$,23,52,4)
    LOOP UNTIL StringIN$ <> ""
    Identification$(15) = StringIN$
    LSET year$ = Identification$(15)
    ERASE TrafCon&
    RESUME 3907
  CASE 33
    ErMsg$(1) = "File " + FileErr$ + " couldn't be found on"
    ErMsg$(2) = "the indicated path, or does not exist."
  CASE 58
    ErMsg$(1) = "The file " + FileErr$ + " already exists "
  CASE 53
    ErMsg$(1) = "File " + FileErr$ + " couldn't be found on"
    ErMsg$(2) = "the indicated path, or does not exist."
  CASE 57
    ErMsg$(1) = "The disk in the indicated or default drive has not"
    ErMsg$(2) = "been formatted, insert a formatted disk to continue"
  CASE 54,64
    ErMsg$(1) = "The file name entered contains invalid characters"
  CASE 61
    ErMsg$(1) = "There isn't enough free space on the drive to carry a"
    ErMsg$(2) = "file option; provide more free space and retry" +_
                " your selection."
  CASE 70,72
    ErMsg$(1) = "The disk is write-protected, or bad diskette"
  CASE 71
    ErMsg$(1) = "The door of a floppy disk drive is open, or"
    ErMsg$(2) = "there is not disk in the indicated drive."
  CASE 76
    ErMsg$(1) = "The path you specified can not be found"
  CASE ELSE
    ErMsg$(1) = "Unidentified Error"
    ErMsg$(2) = "Error Code: " + STR$(ERR) +_
                "  Error Address: " + STR$(ERADR) +_
                "  Error Line: " + STR$(ERL)
    ErMsg$(4) = "Report code and address to PBA developers"
END SELECT
'display the error window
```

```
    CLOSE
    CALL MsgWindow(17,5,6,70,FNATTR%(ColArr%(14),ColArr%(4)),1,0,0)
    ERASE ErMsg$
    SELECT CASE ERL
      CASE 755                  'error on free disk space
        CLOSE # 1
        NoSpace$ = "Y"
        RESUME NEXT
      CASE 17300 TO 17325       'error on data files
        NewName$ = "Y"
        RESUME 17525
      CASE 17330                'existent file
        RESUME NEXT
      CASE 4050                 'error on Help file
        Help$ = "H"
        RESUME NEXT
      CASE ELSE                 'something unexpected went wrong
        RESUME CloseFile
    END SELECT
'------------------------------------------------------------------------------
```

Appendix A.7
PBACAR.BAS

```
'-----------------------------------------------------------------------------
'
'                              PBACAR.BAS
'
'The following program was coded using Turbo BASIC version 1.1
'the following subroutine is used to input the vehicle types to be used
'by the PredictedTraffic subroutine, and to compute the ESALs
'-----------------------------------------------------------------------------
'the following subroutine is used to convert mixed traffic to Equivalent
'Single Axle Loads (ESAL), the procedure is based on the 1986 AASHTO Guide
'for the Design of Pavement Structures, Appendix D
PredictedTraffic:
      $EVENT OFF
      Editing$ = "P"
      GOSUB VehicleTypes
      IF Help$ <> "Q" THEN
        OnHelp$ = "PREDICT"
'print the input screen on the display
        CALL Frame        ' print the main screen
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 24,16,0  :  PRINT " F1 : Help ";
        LOCATE 24,28,0  :  PRINT " F6 : Edit ";
        LOCATE 24,40,0  :  PRINT " ESC : Quit ";
        LOCATE 24,53,0  :  PRINT " F9 : End ";
        CALL ESALscr  ' call the screen for inputing the data
        COLOR ColArr%(15),ColArr%(1)
'create an array to store the data to be used in the PredictedTraffic routine
        EndArray% = VehicleLast% + 10
        DIM DYNAMIC ESAL#(11:EndArray%,1:3)
'This loop prints the different vehicle types on the input screen
        VehTy% = 11  :  Varray% = 1
        EndVehicles% = 20  : pt3% = 11  : Varr% = 1
        FOR Nscr% = 1 TO NxtScr%
          IF Nscr% = NxtScr% THEN EndVehicles% = LaCar% + 10
          FOR VehTy% = 11 TO EndVehicles%
            LOCATE VehTy%,10,0
            PRINT USING "&"; LEFT$(Vehicle$(Varray%),28)
            INCR Varray%
          NEXT VehTy%
          pt1% = 11  :  pt2% = EndVehicles%
'the following loop moves the cursor from column to column in order for the
'user to input the data, then the data are stored in the ESAL$ array
          FOR pt1% = 11 TO pt2%
            pt8% = 1  :  Col% = 41
            COLOR ColArr%(12),ColArr%(15)
            LOCATE pt1%,10,0
            PRINT USING "&"; LEFT$(Vehicle$(Varr%),28)
            FOR ESALdat% = 1 TO 3
3500          LOCATE pt1%,Col%,0
              PRINT SPACE$(10)
              CALL InputDouble(Value#,pt1%,(Col%+1),8)
              ESAL#(pt3%,pt8%) = Value#
```

```
            IF Help$ = "Q" THEN
              ERASE ESAL#,Vehicle$
              EXIT FOR
            END IF
            INCR pt8%   :  INCR Col%,11
          NEXT ESALdat%
          IF Help$ <> "Q" THEN
            COLOR ColArr%(15),ColArr%(1)
            ' restores the color of the vehicle column
            LOCATE pt1%,10,0
            PRINT USING "&"; LEFT$(Vehicle$(Varr%),28)
            INCR pt3%
3800        INCR Varr%
            'the following IF block generates the input screen for the
            'following vehicle types
            IF pt1% = pt2% THEN
              IF Nscr% = NxtScr% THEN EXIT IF
              CALL ESALscr
              COLOR ColArr%(15),ColArr%(1)
            END IF
          END IF
          IF Help$ = "Q" THEN EXIT FOR
        NEXT pt1%
        IF Help$ = "Q" THEN EXIT FOR
      NEXT Nscr%
'the following routine computes the Equivalent Single Axle Loads based on
'the data input above;  the following variables are used to determine the
'length of the pavement cycle life since the construction or rehabilitation
'year to present
      IF Help$ <> "Q" THEN
        DateAC$ = DATE$
        DateAC$ = RIGHT$(DateAC$,4)
        CurrentYear% = VAL(DateAC$)
        'if a new Construction/Rehabilitation year was entered after error 6
        DO
          DO
            DO
3825          IF NewYear$ = "Y" THEN NewYear$ = ""
              ConstructionYear% = VAL(Identification$(15))
              ptano% = CurrentYear% - ConstructionYear%
              'create an array to store the ESALs for each year
              DIM DYNAMIC ESALu!(1:10),TrafCon&(10,1)
              'the following block is used to determine the boundaries
              'used in the TrafCon subroutine
              SELECT CASE ptano%
                CASE 0 TO 10
                  PtCount% = 1   :  PtInit% = 2   :  PtFi% = 1
                CASE 11 TO 15
                  PtCount% = 2   :  PtInit% = 2   :  PtFi% = 1
                CASE 16 TO 20
                  PtCount% = 2   :  PtInit% = 4   :  PtFi% = 2
                CASE 21 TO 25
```

```
                PtCount% = 3  :  PtInit% = 3  :  PtFi% = 1
              CASE 26 TO 30
                PtCount% = 3  :  PtInit% = 6  :  PtFi% = 3
              CASE > 30
                PtCount% = INT(ptano% / 10) + 1
                PtInit% = 2 * PtCount%
                PtFi% = PtCount%
            END SELECT
            m% = 0  ' store the first ESAL on element 0 of the TrafCon array
            n% = PtFi%  :  CALL TrafCon
            'the NewYear$ variable is used if a new Construction/
            'Rehabilitation year was entered
          LOOP UNTIL NewYear$ <> "Y"
          n% = PtInit% : NofMea% = 1
          'the following loop is for ptinitial till before ptano%
          WHILE n% < ptano%
            CALL TrafCon
            IF NewYear$ = "Y" THEN EXIT LOOP
            INCR NofMea%  :  INCR n%,PtCount%
          WEND
        LOOP UNTIL NewYear$ <> "Y"
        'for the last year equal to ptano%
        n% = ptano%   :  CALL TrafCon
      LOOP UNTIL NewYear$ <> "Y"
      INCR NofMea%
      ERASE Vehicle$,ESAL#
      'used in the PresentServiceabilityIndex routine
      PSIpredicted$ = "yes"
    END IF
  END IF
3875  RETURN         ' return to the main data collection routine
'..........................................................................
'the following subroutine converts mixed traffic to ESAL, refer to the 1986
'AASHTO Guide for Design of Pavement Structures, Appendix D
'first, retrieve the data from the ESAL# array
SUB TrafCon
     SHARED EndArray%,ESAL#(),GrowthRate#,NewYear$,TrafCon&(),n%,m%,ESALu!()
     ESALcount% = 11
     TotalESAL& = 0
     WHILE ESALcount% <= EndArray%
       GrowthRate# = ESAL#(ESALcount%,2)
       g# = GrowthRate# / 100
3900   GrowthFactor# = (((1 + g#)^n%) - 1) / g#
       CurrentTraffic# = ESAL#(ESALcount%,1)
       DesignTraffic# = GrowthFactor# * CurrentTraffic# * 365
       ESALfactor# = ESAL#(ESALcount%,3)
3905   DesignESAL& = INT(DesignTraffic# * ESALfactor#)
'the NewYear$ variable is used if a new Construction/Rehabilitation year
'was entered after ERROR 6
3907   IF NewYear$ = "Y" THEN EXIT LOOP
       TotalESAL& = TotalESAL& + DesignESAL&
3910   INCR ESALcount%
```

```
      WEND
      IF NewYear$ <> "Y" THEN
'stores the total Design ESAL for the n th year in the TrafCon array
        ESALu!(m%+1) = TotalESAL&
        TrafCon&(m%,1) = TotalESAL&
        TrafCon&(m%,0) = n%
        INCR m%
3915 END IF
END SUB
'------------------------------------------------------------------------------
SUB ManTraffic
       SHARED Help$
       CALL InputData("T")
END SUB
'------------------------------------------------------------------------------
'the following subroutine is used to input the vehicle types to be used
'by the PredictedTraffic subroutine
VehicleTypes:
      DO
10      NewName$ = ""  :  FileErr$ = ""  :  OnHelp$ = "CAR"
        CALL FileScreen
'create the menu for file creation
50      LOCATE 4,25,0  :  PRINT " Vehicle Types File "
        LOCATE 6,15,0  :  PRINT "1. AASHTO Vehicle Types"
        LOCATE 7,15,0  :  PRINT "2. Open an existent Vehicle-Types File"
        LOCATE 8,15,0  :  PRINT "3. Create a Vehicle-Types File"
        LOCATE 9,15,0  :  PRINT "4. Return to Manual Traffic"
        LOCATE 10,15,0 :  PRINT "   Enter a choice :"
        DO
60        CALL MenuChoice(Choice$,10,36)
          FileChoice% = VAL(Choice$)
          LOCATE 10,37,0
          SELECT CASE FileChoice%
            CASE 1
              FileErr$ = "AASHTO.CAR"
              EXIT LOOP
            CASE 2    ' to open an existing data file to add or modify
              LOCATE 4,25,0  :  PRINT " Open Vehicle-Type File "
              FileSpec$ = " opened"
              EXIT LOOP
            CASE 3    ' to create a brand new data file
              LOCATE 4,25,0  :  PRINT " New Vehicle-Type File "
              FileSpec$ = " created"
              EXIT LOOP
            CASE 4
              Help$ = "Q"
              EXIT LOOP
          END SELECT
        LOOP
        IF Help$ <> "Q" THEN
          SELECT CASE FileErr$
            CASE "AASHTO.CAR"
```

```
                EXIT SELECT
              CASE ELSE
70              COLOR ColArr%(15),ColArr%(1)
                LOCATE 6,13,0
                PRINT "Enter the name of the Vehicle-Type file to be" + FileSpec$
                'enter the name of the file and check the its existence
                FileExt$ = ".CAR" : PerfEqn$ = ""
                Viewing$ = "N"
                CALL FiNaCheck(FileExt$,FileChoice%)
                IF NewName$ = "Y" THEN EXIT SELECT
                Car$ = ""
                CLOSE # 2
            END SELECT
            IF NewName$ <> "Y" THEN
125           OPEN FileErr$ AS # 2 LEN = 50
              'define the length for each field on each record
              FIELD # 2, 50 AS TypeOfVehicle$
              DIM DYNAMIC Vehicle$(1:50)
150           SELECT CASE FileChoice%
                CASE 1
                  'CASE 1 uses the vehicle types given in Appendix D, AASHTO
                  VehicleFirst% = 1   :  VehicleLast% = 15   :  NofVehicles% = 14
                  Vehicle$(1) = "Passenger Cars"
                  Vehicle$(2) = "Buses"
                  Vehicle$(3) = "Panel and Pickup Trucks"
                  Vehicle$(4) = "Other 2-Axle/4-Tire Trucks"
                  Vehicle$(5) = "2-Axle/6-Tire Trucks"
                  Vehicle$(6) = "3 or More Axle Trucks"
                  Vehicle$(7) = "3 Axle Tractor Semi-Trailers"
                  Vehicle$(8) = "4 Axle Tractor Semi-Trailers"
                  Vehicle$(9) = "5+ Axle Tractor Semi-Trailers"
                  Vehicle$(10) = "5 Axle Double Trailers"
                  Vehicle$(11) = "6+ Axle Double Trailers"
                  Vehicle$(12) = "3 Axle Truck-Trailers"
                  Vehicle$(13) = "4 Axle Truck-Trailers"
                  Vehicle$(14) = "5+ Axle Truck-Trailers"
                  Vehicle$(15) = "End of File"
                CASE 2
                  'CASE 2 is for using an already existent file
                  FOR CarCount% = 1 TO 50
                    GET # 2, CarCount%
                    Vehicle$(CarCount%) = TypeOfVehicle$
                    Fin$ = MID$(Vehicle$(CarCount%),1,11)
                    'if the end of the Vehicle$ file is reached then go back to
                    'PredictedTraffic
                    IF Fin$ = "End of File" THEN EXIT FOR
                  NEXT CarCount%
                  NofVehicles% = CarCount% - 1
                CASE 3
                  'CASE 3 is for creating a brand new file
                  'create the screen for entering the vehicle types
                  CALL Frame
```

```
COLOR ColArr%(12),ColArr%(15)
LOCATE 3,34,0  :  PRINT " VEHICLE TYPES "
LOCATE 24,6,0   :  PRINT " F6 : EDIT ";
COLOR ColArr%(15),ColArr%(1)
'enter the number of vehicle types which are to be used
LOCATE 5,20,0  :  PRINT "Input the number of Vehicle Types "
DO
200  LOCATE 6,20,1  :  PRINT "(maximum of 50 vehicle types) : "
  COLOR ColArr%(10),ColArr%(12)
  LOCATE 6,54,0  :  PRINT "     "
  CALL InputInteger(Value%,6,55,2)
  IF Help$ <> "Q" THEN
    NofVehicles% = Value%
    'checking that the number of measurements is < 50 and > 1
    SELECT CASE NofVehicles%
      CASE 1 TO 50
        EXIT LOOP
      CASE ELSE
        COLOR ColArr%(18),ColArr%(1)
    END SELECT
  END IF
LOOP UNTIL Help$ = "Q"
IF Help$ <> "Q" THEN
  COLOR ColArr%(15),ColArr%(1)
  LOCATE 6,20,0  :  PRINT "(maximum of 50 vehicle types) : "
  LOCATE 9,20,0
  PRINT "Enter the Vehicle Types (28 characters maximum)"
  'determine how many screens to use to input all of the types
  CarIn$ = "Y"
  IF NofVehicles% <= 10 THEN
    LaCar% = NofVehicles%
    NxtScr% = 1
  ELSE
    Mcar% = FIX(NofVehicles% / 10)
    IF Mcar% = 5 THEN Mcar% = 4
    LastCar% = NofVehicles% - (Mcar% * 10)
    NxtScr% = Mcar% + 1
  END IF
  EndVehicles% = 10
  CarIn% = 1
  Vcount% = 1
  'create the boxes used to enter the vehicle types
  FOR Nscr% = 1 TO NxtScr%
    IF Nscr% = NxtScr% THEN EndVehicles% = LaCar%
    CarRow% = 11
    FOR Bcar% = 1 TO EndVehicles%
      COLOR ColArr%(15),ColArr%(1)
      LOCATE CarRow%,24,0
      PRINT USING " ## " ;CarIn%
      COLOR ColArr%(12),ColArr%(10)
      LOCATE CarRow%,29,0
      PRINT SPACE$(30)
```

```
              INCR CarRow%
              INCR CarIn%
            NEXT Bcar%
            CarRow% = 11    ' start at row number 11
            'enter the vehicle types
            FOR InVehi% = 1 TO EndVehicles%
              COLOR ColArr%(12),ColArr%(15)
225           LOCATE CarRow%,24,0  :  PRINT USING " ## "; Vcount%
              LOCATE CarRow%,29,0  :  PRINT STRING$(30,0)
              CALL InputString$(StringIN$,CarRow%,30,28)
              IF Help$ <> "Q" THEN
                Vehicle$(Vcount%) = StringIN$
                COLOR ColArr%(15),ColArr%(1)
                LOCATE CarRow%,24,0  :  PRINT USING " ## "; Vcount%
                COLOR ColArr%(12),ColArr%(10)
                LOCATE CarRow%,29,0  :  PRINT " ";
                PRINT USING "&"; LEFT$(Vehicle$(Vcount%),28)
                INCR Vcount%  :  INCR CarRow%
              END IF
              IF Help$ = "Q" THEN EXIT FOR
            NEXT InVehi%
            IF Help$ <> "Q" THEN
              COLOR ColArr%(15),ColArr%(1)
              'clean the screen before continuing with the next screen
              FOR row% = 11 TO 21
                LOCATE row%,25,0
                PRINT SPACE$(35)
              NEXT row%
            END IF
            IF Help$ = "Q" THEN EXIT FOR
          NEXT Nscr%
          IF Help$ <> "Q" THEN
            Vehicle$(Vcount%) = "End of File"
            CarIn$ = ""
          ELSE
            ERASE Vehicle$
          END IF
        END IF
250   END SELECT
      IF Help$ <> "Q" THEN
        'save the different vehicle types in Vehicle$ before continuing
        IF FileChoice% <> 2 THEN
300       FOR CarCount% = 1 TO 50
            LSET TypeOfVehicle$ = Vehicle$(CarCount%)
            PUT # 2, CarCount%
          NEXT CarCount%
        END IF
        'NxtScr% gives the number of screens used to input the data
350     NxtScr% = FIX(NofVehicles% / 10) + 1
        LaCar% = NofVehicles% - ((NxtScr% - 1) * 10)
        VehicleFirst% = 1
        VehicleLast% = NofVehicles%
```

```
          CLOSE # 2
        END IF
      END IF
    END IF
    IF Help$ = "Q" THEN EXIT LOOP
  LOOP UNTIL NewName$ <> "Y"
375  RETURN
'-------------------------------------------------------------------------------
'this subroutine is used to edit values on the main data collection routine
SUB CarEdit
      SHARED Help$,VehicleChange$,CarRow%,Nscr%,Vehicle$(),ColArr%()
      SHARED CarModify%,LastVehicle%,FirstCar%
      Help$ = "E"
      CALL EditScreen
      VehicleChange$ = ""
      LastVehicle% = CarRow%  :  Cedit% = 11
      IF LastVehicle% <> 11 THEN
        CarModify% = ((Nscr% - 1) * 10) + 1  :  FirstCar% = CarModify%
        DO
11000     COLOR ColArr%(10),ColArr%(12)
          LOCATE Cedit%,29,0  :  PRINT " ";
          PRINT USING "&" ; LEFT$(Vehicle$(CarModify%),28)
          DO
11100       DO : LOOP UNTIL INSTAT
            Vchange$ = INKEY$
            SELECT CASE Vchange$
              CASE CHR$(13)
                'if the ENTER key is pressed then input the new value
                LOCATE Cedit%,29,0  :  PRINT SPACE$(30)
                CALL EditString$(EdStr$,Cedit%,30,28)
                Vehicle$(CarModify%) = EdStr$
                EXIT LOOP
              CASE CHR$(27)
                'if the ESC key is pressed then return to the input screen
                EXIT LOOP
              CASE ELSE
                SELECT CASE MID$(Vchange$,2,1)
                  CASE CHR$(72),CHR$(80)
                    CALL Vabajo
                END SELECT
            END SELECT
          LOOP
          IF Vchange$ = CHR$(27) THEN EXIT LOOP
        LOOP
        COLOR ColArr%(12),ColArr%(10)
        LOCATE Cedit%,29,0  :  PRINT " ";
        PRINT USING "&"; LEFT$(Vehicle$(CarModify%),28)
      END IF
      CALL ClEdSc
      COLOR ColArr%(12),ColArr%(10)
      LOCATE 3,34,0  :  PRINT " VEHICLE TYPES "
      COLOR ColArr%(12),ColArr%(15)
```

```
      LOCATE 24,6,0   :   PRINT " F6 : EDIT ";
END SUB
'.............................................................................
'the following routine moves the cursor vertically when using the
'CarEdit routine
SUB Vabajo
      SHARED Cedit%,Vehicle$(),CarModify%,LastVehicle%,FirstCar%,ColArr%()
      COLOR ColArr%(12),ColArr%(10)
      LOCATE Cedit%,29,0   :   PRINT " ";
      PRINT USING "&"; LEFT$(Vehicle$(CarModify%),28)
      INCR Cedit%   :   INCR CarModify%
      IF Cedit% = LastVehicle% THEN
        Cedit% = 11
        CarModify% = FirstCar%
      END IF
      COLOR ColArr%(10),ColArr%(12)
      LOCATE Cedit%,29,0   :   PRINT " ";
      PRINT USING "&"; LEFT$(Vehicle$(CarModify%),28)
END SUB
'.............................................................................
```

Appendix A.8
PBAFILE.BAS

```
'-----------------------------------------------------------------------
'
'                          PBAFILE.BAS
'
'the following program was coded using Turbo BASIC version 1.1
'this file contains the subroutines used during the execution of the
'main file "PBAASHTO.BAS" to enter the name of the data file to use
'-----------------------------------------------------------------------
'turn key trapping off
      $EVENT OFF
'-----------------------------------------------------------------------
SUB FiNaCheck(FileExt$,FileChoice%)
      SHARED ColArr%(),Editing$,OnHelp$,FileName$,FileEqn$,FileErr$,Viewing$
      SHARED Help$,FileDat$,FileCar$,NewName$
      LOCATE 7,13,0 : PRINT "You can add a disk drive address (eg. A:), or a"
      LOCATE 8,13,0 : PRINT "directory path (eg. C:\PBA\ ). DO NOT ADD A FILE"
      LOCATE 9,13,0 : PRINT "EXTENSION .  (40 characters maximum) : "
      LOCATE 10,15,0 : PRINT SPACE$(50)
17200 LOCATE 10,15,0
      COLOR ColArr%(12),ColArr%(15)
      PRINT SPACE$(41)
      Editing$ = "F"
      CALL InputString(StringIN$,10,16,40)
      IF Help$ <> "Q" THEN
        FileName$ = UCASE$(StringIN$)
        LOCATE 10,16,0
        Drive% = 0  ' to check free memory on disk
        'detect disk drive being used
        CharacterCheck$ = MID$(FileName$,2,1)
        LetterCheck% = INSTR(1,":\",CharacterCheck$)
        IF LetterCheck% <> 0 THEN
          Drive$ = MID$(FileName$,1,1)
          IF Drive$ = "A" THEN Drive% = 1
          IF Drive$ = "B" THEN Drive% = 2
          IF Drive$ = "C" THEN Drive% = 3
        END IF
        'find out which routine it is coming from
        SELECT CASE FileExt$
          CASE ".EQN"
            FileEqn$ = FileName$ + ".EQN"
            FileErr$ = FileEqn$
17300       OPEN FileEqn$ AS # 4 LEN = 472
            'determine if at least 12K bytes is available in the default
            'or indicated disk
            NofRecords& = INT(FNbytes(Drive%) / 472) - 15
17305       IF NofRecords& < 15 THEN ERROR 61
            Test& = LOF(4)  :  PerfEqn$ = "Y"
          CASE ".DAT"
            FileDat$ = FileName$ + ".DAT"
            FileErr$ = FileDat$
17310       OPEN FileDat$ AS # 1 LEN = 197
            'determine if at least 1K bytes is available in the default
```

```
                'or indicated disk
                NofRecords& = INT(FNbytes(Drive%) / 197) - 6
17315           IF NofRecords& < 5 THEN ERROR 61
                Test& = LOF(1)  :  FileMain$ = "Y"
              CASE ".CAR"
                FileCar$ = FileName$ + ".CAR"
                FileErr$ = FileCar$
17320           OPEN FileCar$ AS # 2 LEN = 50
                'determine if at least 3K bytes is available in the default
                'or indicated disk
                NofRecords& = INT(FNbytes(Drive%) / 50) - 50
17325           IF NofRecords& < 50 THEN ERROR 61
                Test& = LOF(2)  :  Car$ = "Y"
            END SELECT
            'the following subroutine checks the existence of the file
            IF Test& = 0 THEN
              IF FileChoice% = 2 OR FileChoice% = 3 THEN
                CALL CloseFile(PerfEqn$,FileMain$,Car$)
                KILL FileErr$
                NewName$ = "Y"
17330           ERROR 33
              END IF
            END IF
            COLOR ColArr%(15),ColArr%(1)
            'if using an existent .CAR data file
            IF Car$ = "Y" AND Test& > 0 THEN
              EXIT IF
            ELSE
              IF Test& > 0 AND Viewing$ = "N" THEN
                FOR ClearScr% = 13 TO 18
                  LOCATE ClearScr%,11,0
                  PRINT SPACE$(58)
                NEXT ClearScr%
                COLOR ColArr%(12),ColArr%(15)
                LOCATE 14,12,0
                PRINT " " + FileErr$ + " ";
                COLOR ColArr%(15),ColArr%(1)
                PRINT " already exists."
                LOCATE 16,12,0 : PRINT "Do you want to overwrite/delete it (y/n) ?: "
                LOCATE 17,12,0 : PRINT "Do you want to add new information (y/n) ?: "
                COLOR ColArr%(12),ColArr%(15)
                DO
                  CALL MenuChoice(Choice$,16,57)
                  OverWrite$ = Choice$
                  SELECT CASE OverWrite$
                    CASE "N","n"  'if the user does not want to overwrite FileName$
                      DO
                        CALL MenuChoice(Choice$,17,57)
                        add$ = Choice$
                        SELECT CASE add$
                          CASE "N","n"
                            CALL CloseFile(PerfEqn$,FileMain$,Car$)
```

```
                    CALL CleanScr
                    NewName$ = "Y"
                    EXIT LOOP
                  CASE "Y","y"   ' if the user is to add data
                    EXIT LOOP
                END SELECT
              LOOP
              EXIT IF
            CASE "Y","y"
              CALL CloseFile(PerfEqn$,FileMain$,Car$)
              COLOR ColArr%(15),ColArr%(1)
              FOR warn% = 13 TO 18 STEP 1  ' display the warning screen
                LOCATE warn%,11,0  :  PRINT SPACE$(56)
              NEXT warn%
              LOCATE 14,12,0 :  PRINT "The data stored in ";
              COLOR ColArr%(12),ColArr%(15)
              PRINT " " + FileErr$ + " "
              COLOR ColArr%(15),ColArr%(1)
              LOCATE 15,12,0 :  PRINT "are to be deleted"
              LOCATE 16,12,0 :  PRINT "Enter 'y' to continue, 'n' to cancel"
              COLOR ColArr%(12),ColArr%(15)
              DO
                CALL MenuChoice(Choice$,16,52)
                warned$ = Choice$
                SELECT CASE warned$
                  CASE "Y","y"            ' erase the FileDat$ file
                    LOCATE 16,51,0
                    KILL FileErr$
                    EXIT LOOP
                  CASE "N","n"
                    COLOR ColArr%(15),ColArr%(1)
                    NewName$ = "Y"
                    EXIT LOOP
                END SELECT
              LOOP
              EXIT IF
          END SELECT
        LOOP
      END IF
    END IF
  ELSE
    NewName$ = "Y"
  END IF
17525 OnHelp$ = ""
END SUB
'.........................................................................
'close the opened file if needed
SUB CloseFile(PerfEqn$,FileMain$,Car$)
    IF PerfEqn$ = "Y" THEN CLOSE # 4
    IF FileMain$ = "Y" THEN CLOSE # 1
    IF Car$ = "Y" THEN CLOSE # 2
END SUB
```

```
'..........................................................................
'the following function determines the amount of free space available in the
'default or indicated disk drive by invoking DOS Service 36H through DOS
'Interrupt 21H
DEF FNbytes(Drive%)
    REG 4,Drive%  'pass drive code to DL:DX (0=default,1=A:,2=B:,3=C:,etc.)
    REG 1,&H3600  ' pass function code to AH:AX
    CALL INTERRUPT &H21
    FNbytes = CSNG(REG(2))*REG(3)*REG(1) 'determine free space (AX*BX*CX)
END DEF
'---------------------------------------------------------------------------
SUB FileScreen
      SHARED ColArr%()
18250 COLOR ColArr%(15),ColArr%(1)
      CLS
      LOCATE 3,10,0  :  PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      LOCATE 4,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      LOCATE 5,10,0  :  PRINT CHR$(195) + STRING$(58,196) + CHR$(180)
      FOR Fb% = 6 TO 10
        LOCATE Fb%,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      NEXT Fb%
      LOCATE 11,10,0 :  PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
      LOCATE 12,10,0 :  PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      FOR Fb% = 13 TO 18
        LOCATE Fb%,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      NEXT Fb%
      LOCATE 19,10,0 :  PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
END SUB
'---------------------------------------------------------------------------
SUB CleanScr
      SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 4,11,0  :  PRINT SPACE$(58)
      FOR Fb% = 6 TO 10
        LOCATE Fb%,11,0
        PRINT SPACE$(58)
      NEXT Fb%
      FOR Fb% = 13 TO 18
        LOCATE Fb%,11,0
        PRINT SPACE$(58)
      NEXT Fb%
      FOR Fb% = 21 TO 23
        LOCATE Fb%,11,0
        PRINT SPACE$(58)
      NEXT Fb%
      LOCATE 24,11,0 :  PRINT SPACE$(58);
END SUB
'---------------------------------------------------------------------------
'the following subroutine is used to enter different types of data
SUB InputData(InDat$)
18000 SHARED ColArr%(),HighwayNo$,SubEqID$,Neck$,NoVar%,Editing$,Help$
      SHARED VarNameX$,UnitsX$,VarX$,Col%,Identification$(),NoData%,Box%
```

```
      SHARED Dat!(),ESALu!(),PSImanual$,OnHelp$,DatBck$
      CALL Frame
      DatBck$ = InDat$
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 24,16,0  :  PRINT " F1 : Help ";
      LOCATE 24,28,0  :  PRINT " F4 : Edit ";
      LOCATE 24,40,0  :  PRINT " ESC : Quit ";
      LOCATE 24,53,0  :  PRINT " F9 : End ";
      COLOR ColArr%(15),ColArr%(1)
      Editing$ = "M"  :
18100 SELECT CASE InDat$
        CASE "E"
          LOCATE 3,20,0 : PRINT USING "Highway ID  : & "; LEFT$(HighwayNo$,10);
          LOCATE 4,20,0 : PRINT USING "Equation ID : & "; LEFT$(SubEqID$,10);
          LOCATE 5,20,0 : PRINT USING "Variable    : & "; LEFT$(Neck$,25)
          LOCATE 6,20,0 : PRINT USING "No. Variable: # "; NoVar%
          CALL DataFrame(39,20,19)
          COLOR ColArr%(12),ColArr%(15)
          LOCATE 10,26,0 : PRINT " YEAR "
          LOCATE 10,45,0 : PRINT " MEASURE "
          OnHelp$ = "USER"
        CASE "T"
          OnHelp$ = "Xvariable"
          LOCATE 5,20,0  :  PRINT "Enter the variable name and units:"
          LOCATE 6,25,0  :  PRINT "Name:"
          LOCATE 7,25,0  :  PRINT "Units:"
          LOCATE 8,25,0  :  PRINT "Variable ID:"
          COLOR ColArr%(12),ColArr%(15)
          LOCATE 3,30,0  :  PRINT " PREDICTION VALUES "
          LOCATE 4,17,0
          PRINT " Independent Variable of Predicted Performance "
          LOCATE 6,32,0  :  PRINT SPACE$(30)
          LOCATE 7,32,0  :  PRINT SPACE$(30)
          LOCATE 8,38,0  :  PRINT SPACE$(7)
          CALL InputString(StringIN$,6,33,25)
18200     IF Help$ <> "Q" THEN
            VarNameX$ = StringIN$
            CALL InputString(StringIN$,7,33,25)
            IF Help$ <> "Q" THEN
              UnitsX$ = StringIN$
              CALL InputString(StringIN$,8,39,5)
              IF Help$ <> "Q" THEN
                VarX$ = StringIN$
                COLOR ColArr%(15),ColArr%(1)
                FOR I% = 5 TO 7
                  LOCATE I%,20,0  :  PRINT SPACE$(50)
                NEXT I%
              END IF
            END IF
            IF Help$ <> "Q" THEN
              CALL DataFrame(39,20,19)
              COLOR ColArr%(12),ColArr%(15)
```

```
            Col% = LEN(Identification$(8)) + 15
            Col% = (80 - Col%) / 2
            LOCATE 5,Col%,0
            PRINT USING " Highway ID : & "; Identification$(8);
            LOCATE 10,28,0 :  PRINT "    Year          " + VarX$ + "      "
          END IF
        END IF
      END SELECT
18300 IF Help$ <> "Q" THEN
        LOCATE 7,20,0  : COLOR ColArr%(15),ColArr%(1)
        PRINT "Input the number of measurements "
        DO
          LOCATE 8,20,0  :  PRINT "(maximum of 10 measurements) : "
          COLOR ColArr%(12),ColArr%(15)
          LOCATE 8,56,0  :  PRINT SPACE$(4)
          CALL InputInteger(Value%,8,57,2)
          IF Help$ = "Q" THEN EXIT LOOP
          NoData% = Value%
          'checking that the number of measurements is < 10 and > 1
          IF NoData% < 1 OR NoData% > 10 THEN
            COLOR ColArr%(18),ColArr%(1)
          ELSE
            EXIT LOOP
          END IF
        LOOP
      END IF
18400 IF Help$ <> "Q" THEN
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 8,20,0  :  PRINT "(maximum of 10 measurements) : "
        Box% = 1
        'the following loop turns on the boxes for inputing the data
        WHILE Box% <= NoData%
          COLOR ColArr%(15),ColArr%(1)
          LOCATE (Box%+11),22,0  :  PRINT Box%
          COLOR ColArr%(12),ColArr%(10)
          LOCATE (Box%+11),25,0  :  PRINT SPACE$(15)
          LOCATE (Box%+11),44,0  :  PRINT SPACE$(15)
          INCR Box%
        WEND
        'the following loop requests the data, creates an array of 10 rows and 2
        'columns to store the data before saving them in the data base file
        DIM DYNAMIC Dat!(1:10,1:2)
        Box% = 1
18500   WHILE Box% <= NoData%
          COLOR ColArr%(12),ColArr%(15)
          DO
            LOCATE (Box%+11),22,0  :  PRINT Box%
            LOCATE (Box%+11),25,0  :  PRINT SPACE$(15)
            RowDat% = Box% + 11
            CALL InputReal(Value!,RowDat%,27,13)
          LOOP UNTIL Help$ <> "E"
          IF Help$ <> "Q" THEN
```

```
            Dat!(Box%,1) = Value!
            DO
              LOCATE (Box%+11),44,0  : PRINT SPACE$(15)
              RowDat% = Box% + 11
              CALL InputReal(Value!,RowDat%,46,13)
            LOOP UNTIL Help$ <> "E"
18600       IF Help$ <> "Q" THEN
              Dat!(Box%,2) = Value!
              COLOR ColArr%(15),ColArr%(1)
              LOCATE (Box%+11),22,0  :   PRINT Box%
              INCR Box%
            END IF
          END IF
          IF Help$ = "Q" THEN
            ERASE Dat!
            EXIT LOOP
          END IF
        WEND
        IF Help$ <> "Q" THEN
          SELECT CASE InDat$
            CASE "T"
              DIM DYNAMIC ESALu!(0:10)
              FOR I% = 1 TO 10
                ESALu!(I%) = Dat!(I%,2)
              NEXT I%
              PSImanual$ = "yes"   'used in PresentServiceabilityIndex routine
              ERASE Dat!
            CASE ELSE
              Editing$ = "E"  'flag used for UserEquation routine
          END SELECT
        END IF
18700 END IF
END SUB
'.....................................................................
'create the frame for inputing data
SUB DataFrame(Bound%,Edge%,Cln%)
      SHARED ColArr%()
      COLOR ColArr%(14),ColArr%(1)
      LOCATE 9,Edge%,0
      PRINT CHR$(218) + STRING$(Bound%,196) + CHR$(191)
      LOCATE 10,Edge%,0 : PRINT CHR$(179) + SPACE$(Bound%) + CHR$(179)
      LOCATE 11,Edge%,0
      PRINT CHR$(195)+STRING$(Cln%,196)+CHR$(194)+STRING$(Cln%,196)+CHR$(180)
      FOR I% = 12 TO 21
        LOCATE I%,Edge%,0
        PRINT CHR$(179) + SPACE$(Cln%) + CHR$(179) + SPACE$(Cln%) + CHR$(179)
      NEXT I%
      LOCATE 22,Edge%,0
      PRINT CHR$(192)+STRING$(Cln%,196)+CHR$(193)+STRING$(Cln%,196)+CHR$(217)
END SUB
'.....................................................................
'display the error screen
```

```
SUB ErrorScreen
18750 SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      LOCATE 20,10,0 :  PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
      FOR Fb% = 21 TO 23
        LOCATE Fb%,10,0  :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
      NEXT Fb%
      LOCATE 24,10,0 :  PRINT CHR$(179) + STRING$(58,0) + CHR$(179);
      LOCATE 25,10,0 :  PRINT CHR$(192) + STRING$(58,196) + CHR$(217);
      COLOR ColArr%(16),ColArr%(10)
      LOCATE 21,36,0 :  PRINT " ERROR "
      COLOR ColArr%(12),ColArr%(10)
      LOCATE 24,28,0 :  PRINT " press ESC to continue ";
      COLOR ColArr%(12),ColArr%(15)
END SUB
'-------------------------------------------------------------------------
'create the data collection screen for the identification data
SUB Frame
18800 SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      CLS
      LOCATE 1,2,0   :  PRINT CHR$(201) + STRING$(74,205) + CHR$(187)
      FOR I% = 2 TO 22
        LOCATE I%,2,0
        PRINT CHR$(186) + STRING$(74,0) + CHR$(186)
      NEXT I%
      LOCATE 23,2,0 :  PRINT CHR$(204) + STRING$(74,205) + CHR$(185)
      LOCATE 24,2,0 :  PRINT CHR$(186) + STRING$(74,0) + CHR$(186);
      LOCATE 25,2,0 :  PRINT CHR$(200) + STRING$(74,205) + CHR$(188);
END SUB
'-------------------------------------------------------------------------
'routine used to enter an integer value
SUB InputInteger(Value%,RowDat%,ColDat%,MaxNoChr%)
18850 SHARED Help$
      Value% = 0
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
      IF Help$ <> "Q" THEN Value% = CINT(VAL(StringIN$))
END SUB
'-------------------------------------------------------------------------
'routine used to enter a long integer value
SUB InputLong(Value&,RowDat%,ColDat%,MaxNoChr%)
18860 SHARED Help$
      Value& = 0
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
      IF Help$ <> "Q" THEN Value& = CLNG(VAL(StringIN$))
END SUB
'-------------------------------------------------------------------------
'routine used to enter a real value
SUB InputReal(Value!,RowDat%,ColDat%,MaxNoChr%)
18870 SHARED Help$
      Value! = 0
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
```

```
      IF Help$ <> "Q" THEN Value! = CSNG(VAL(StringIN$))
END SUB
'------------------------------------------------------------------------
'routine used to enter a string
SUB InputDouble(Value#,RowDat%,ColDat%,MaxNoChr%)
18880 SHARED Help$
      Value# = 0
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
      IF Help$ <> "Q" THEN Value# = CDBL(VAL(StringIN$))
END SUB
'------------------------------------------------------------------------
'routine used to enter data
SUB InputString(StringIN$,RowDat%,ColDat%,MaxNoChr%)
18890 SHARED Help$
      CALL EnterData(RowDat%,ColDat%,StringIN$,MaxNoChr%)
END SUB
'------------------------------------------------------------------------
'this routine is used to select a menu option
SUB MenuChoice(Choice$,RowIn%,ColIn%)
18900 SHARED Help$,ColArr%(),OnHelp$
      LOCAL ExitSub$
      COLOR ColArr%(12),ColArr%(15)
      DO
        LOCATE RowIn%,ColIn%-1,0  :  PRINT SPACE$(3);
        Help$ = "N"
        LOCATE RowIn%,ColIn%,1
        DO  :  LOOP UNTIL INSTAT
        IF Help$ <> "Y" THEN
          Choice$ = INKEY$
          IF Choice$ = CHR$(27) THEN
              ExitSub$ = CHR$(13)
              Help$ = "Q"
          ELSEIF MID$(Choice$,2,1) = CHR$(59) THEN
            CALL PBAhelp(OnHelp$)
          ELSEIF Choice$ <> CHR$(13) OR Choice$ <> CHR$(8) THEN
            LOCATE RowIn%,ColIn%,0  :  PRINT Choice$
            LOCATE RowIn%,ColIn%+1,1
            DO :  LOOP UNTIL INSTAT
            ExitSub$ = INKEY$
          END IF
        END IF
      LOOP UNTIL ExitSub$ = CHR$(13)
END SUB
'........................................................................
'this routine enters a string of up to MaxNoChr% characters
SUB EnterData(DataRow%,DataCol%,InString$,MaxNoChr%)
18950 SHARED Help$,Editing$,CarIn$,OnHelp$,EditEq$
      LOCAL LetPos%,NoChr%,Lines%,StartCol%
      InString$ = STRING$(MaxNoChr%,0) : LetPos% = 1
      StartCol% = DataCol%  :  Help$ = ""
      NoChr% = 1  :  DataCol% = StartCol%
      DO WHILE NoChr% <= MaxNoChr%
```

```
      Help$ = "N"
      DO
        LOCATE DataRow%,DataCol%,1
      LOOP UNTIL INSTAT
      Letter$ = INKEY$
      SELECT CASE Letter$
        CASE CHR$(27)
          Help$ = "Q"
        CASE CHR$(13)
          EXIT LOOP
        CASE CHR$(8)
          IF DataCol% > StartCol% THEN    'avoid backspacing
            DECR DataCol% : DECR NoChr%  :  DECR LetPos%
            LOCATE DataRow%,DataCol%,0  :  PRINT SPACE$(1)
            MID$(InString$,LetPos%,1) = CHR$(0)
          END IF
18975   CASE ELSE
          SELECT CASE MID$(Letter$,2,1)
            CASE CHR$(59)
              CALL PBAhelp(OnHelp$)
            CASE CHR$(60)
              IF Editing$ = "I" THEN CALL IDdatEd
              IF Editing$ = "E" THEN
                CALL NeEqE
                Help$ = "E"
              END IF
            CASE CHR$(61)
              IF Editing$ = "Eq" THEN
                Help$ = "Q"
                EditEq$ = "Y"
              END IF
            CASE CHR$(62)
              IF Editing$ = "M" THEN CALL ManTrEd
            CASE CHR$(64)
              IF Editing$ = "P" THEN
                IF CarIn$ = "Y" THEN
                  CALL CarEdit
                ELSE
                  CALL PreTrEd
                END IF
              END IF
            CASE CHR$(66)
              IF Editing$ = "D" THEN CALL DefEdit
              IF Editing$ = "A" THEN
                CALL StrDatEd
                IF Help$ = "C" THEN Help$ = "E"
              END IF
            CASE CHR$(67)
              GOTO CloseFile:
            CASE ELSE
              IF LEN(Letter$) = 1 THEN
                LOCATE DataRow%,DataCol%,0
```

```
                PRINT Letter$
                MID$(InString$,LetPos%,1) = Letter$
                INCR DataCol% : INCR NoChr%  :  INCR LetPos%
              END IF
          END SELECT
      END SELECT
      IF INSTR(1,"EQ",Help$) <> 0 THEN EXIT LOOP
    LOOP
18995 IF INSTR(1,"EQ",Help$) = 0 THEN
      InString$ = LEFT$(InString$,LetPos%-1)
    END IF
END SUB
'.......................................................................
'display the equation screen
SUB EqnScr(FileName$)
39340 SHARED ColArr%(),FileEqn$
      COLOR ColArr%(14),ColArr%(1)
      CLS
      LOCATE 1,1,0 : PRINT CHR$(218) + STRING$(78,196) + CHR$(191)
      FOR Frame% = 2 TO 22
        LOCATE Frame%,1,0
        PRINT CHR$(179) + SPACE$(78) + CHR$(179)
      NEXT Frame%
39360 LOCATE 7,1,0  : PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
      IF RIGHT$(FileName$,4) = ".EQN" THEN
        LOCATE 19,1,0
        PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
      END IF
      LOCATE 23,1,0 : PRINT CHR$(195) + STRING$(78,196) + CHR$(180);
      LOCATE 24,1,0 : PRINT CHR$(179) + SPACE$(78) + CHR$(179);
      LOCATE 25,1,0 : PRINT CHR$(192) + STRING$(78,196) + CHR$(217);
      COLOR ColArr%(12),ColArr%(15)
      Col% = ((80 - LEN(FileName$))/2) - 1
39380 LOCATE 1,Col%,0 : PRINT " " + UCASE$(FileName$) + " "
      LOCATE 24,24,0  : PRINT " ENTER : Continue ";
      LOCATE 24,43,0  : PRINT  " ESC : Quit ";
END SUB
'.......................................................................
'clean equation screen
SUB ClScr(FileName$)
39400 SHARED ColArr%()
      COLOR ColArr%(14),ColArr%(1)
      FOR Frame% = 2 TO 22
        LOCATE Frame%,2,0
        PRINT SPACE$(78)
      NEXT Frame%
      LOCATE 7,1,0  : PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
      IF RIGHT$(FileName$,4) = ".EQN" THEN
        LOCATE 19,1,0
        PRINT CHR$(195) + STRING$(78,196) + CHR$(180)
39420 END IF
END SUB
```

Appendix A.9
PBAEDIT.BAS

```
'------------------------------------------------------------------------
'
'                              PBAEDIT.BAS
'
'the following program was coded using Turbo BASIC version 1.1
'this file contains the subroutines used during the execution of the
'main file "PBAASHTO.BAS" to change data on the different input screens
'------------------------------------------------------------------------
'turn key trapping off
      $EVENT OFF
'------------------------------------------------------------------------
'this subroutine changes the values on the main data collection routine
SUB IDdatEd
8005  SHARED RowID%,Identification$(),ColArr%(),IDrow%,IDlast%,Help$
      CALL EditScreen
      Help$ = "C"
      IDlast% = RowID% : IDrow% = 8
      DO
        COLOR ColArr%(10),ColArr%(12)
        LOCATE IDrow%,48,0
        PRINT "   "; : PRINT USING "\           \" ;Identification$(IDrow%)
        DO
8010      WHILE NOT INSTAT  :  WEND
          IDedit$ = INKEY$
'if the ENTER key is pressed then input the new value
          SELECT CASE IDedit$
            CASE CHR$(13)
              LOCATE IDrow%,48,0  :  PRINT SPACE$(15)
              CALL EditString(EdStr$,IDrow%,51,13)
              Identification$(IDrow%) = EdStr$
              EXIT LOOP
'if the ESC key is pressed then return to the input screen
            CASE CHR$(27)
              EXIT LOOP
            CASE ELSE
              SELECT CASE MID$(IDedit$,2,1)
                CASE CHR$(72),CHR$(80)
                  CALL IDown
              END SELECT
          END SELECT
        LOOP
        IF IDedit$ = CHR$(27) THEN EXIT LOOP
      LOOP
      COLOR ColArr%(12),ColArr%(15)
      LOCATE IDrow%,48,0
      CALL IDprint
      CALL ClEdSc
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 3,23,0  :  PRINT " PREDICTED SERVICEABILITY MODULE "
      LOCATE 4,34,0  :  PRINT " DATA BASE "
      LOCATE 24,17,0 :  PRINT " F1 : Help ";
      LOCATE 24,29,0 :  PRINT " F2 : Edit ";
```

```
      LOCATE 24,41,0 :  PRINT " F9 : End ";
      LOCATE 24,52,0 :  PRINT " ESC : Menu ";
END SUB
'.............................................................................
'the following routine moves the cursor vertically when using IDdatEd routine
SUB IDown
8015  SHARED ColArr%(),IDrow%,IDlast%
      COLOR ColArr%(12),ColArr%(15)
      LOCATE IDrow%,48,0
      CALL IDprint
      INCR IDrow%
      IF IDrow% >= IDlast% OR IDrow% > 16 THEN IDrow% = 8
      COLOR ColArr%(10),ColArr%(12)
      LOCATE IDrow%,48,0
      CALL IDprint
END SUB
'.............................................................................
'this subroutine is used to print the data on the boxes
SUB IDprint
8025  SHARED IDrow%,Identification$()
      PRINT USING "   \           \"; Identification$(IDrow%)
END SUB
'-----------------------------------------------------------------------------
'this subroutine edits values for the InputData subroutine
SUB ManTrEd
8035 SHARED DatBck$,Help$,Box%,NoData%,Dat!(),Identification$(),Head$,HighwayNo$
     SHARED SubEqID$,Neck$,ColArr%(),Col%,MTedit%,MtR%,MtC%,DaR%,DaC%,MtLaR%
     SHARED ColH%,ChT$,NoVar%
     CALL EditScreen
     IF DatBck$ = "T" THEN
       COLOR ColArr%(15),ColArr%(1)
       LOCATE 24,11,0  :  PRINT "Switch to PREDICTED TRAFFIC input (y/n) ?";
       COLOR ColArr%(12),ColArr%(15)
       DO
         LOCATE 24,55,0  :  PRINT "   ";
         LOCATE 24,56,1
         DO  :  LOOP UNTIL INSTAT
         Switch$ = UCASE$(INKEY$)
         SELECT CASE Switch$
           CASE "Y","N"
             IF Switch$ = "Y" THEN
               Help$ = "Q"
               ChT$ = "Y"
             END IF
             EXIT LOOP
         END SELECT
       LOOP
     END IF
     IF Help$ <> "Q" THEN
       Help$ = "E"
       COLOR ColArr%(15),ColArr%(1)
       LOCATE 24,11,0  :  PRINT SPACE$(50);
```

```
      MtLaR% = (Box%+11)  :  MtR% = 8       ' controlling variables
      MtC% = 56  :  DaR% = 1   :  DaC% = 1   ' for the MoDo routine
      DO
8040    CALL PrTra(MtR%,DaC%,ColArr%(10),ColArr%(12))
        DO
8045      DO  :  LOOP UNTIL INSTAT
          MTedit$ = INKEY$   ' after pressing RETURN the data is to be changed
          SELECT CASE MTedit$
            CASE CHR$(13)
'the following IF block is used to change the number of boxes if needed
              IF MtR% = 8 THEN
                DO
8047              LOCATE MtR%,MtC%,0  :  PRINT "    "
                  LOCATE MtR%,(MtC%+2),1
                  CALL EditInteger(EdValue%,MtR%,(MtC%+2),2)
                  NewNoM% = EdValue%
                  SELECT CASE NewNoM%
                    CASE 1 TO 10
'the following IF block is used to increase the number of boxes
                      IF NewNoM% > NoData% THEN
                        NewM% = NoData% + 1
                        WHILE NewM% <= NewNoM%    ' display the new boxes
                          COLOR ColArr%(15),ColArr%(1)
                          LOCATE (NewM%+11),22,0  :  PRINT NewM%
                          COLOR ColArr%(12),ColArr%(10)
                          LOCATE (NewM%+11),25,0  :  PRINT SPACE$(15)
                          LOCATE (NewM%+11),44,0  :  PRINT SPACE$(15)
                          INCR NewM%
                        WEND
                        COLOR ColArr%(12),ColArr%(15)
                        NoData% = NewNoM%    ' change the old variable
                        LOCATE 8,56,0  :  PRINT USING " ## "; NoData%
                        EXIT LOOP    ' return to the main data input screen
                      'if the number is the same or smaller then don't change
                      ELSE
                        COLOR ColArr%(12),ColArr%(15)
                        LOCATE 8,56,0  :  PRINT USING " ## "; NoData%
                        EXIT LOOP
                      END IF
                  END SELECT
                LOOP
              ELSE
                LOCATE MtR%,MtC%,0  :  PRINT SPACE$(15)
                CALL EditReal(EdValue!,MtR%,(MtC%+2),11)
                Dat!(DaR%,DaC%) = EdValue!
                EXIT LOOP
              END IF
            CASE CHR$(27)
              EXIT LOOP
            CASE ELSE
              SELECT CASE MID$(MTedit$,2,1)
                CASE CHR$(72),CHR$(80)
```

```
                CALL MoDo
              CASE CHR$(75),CHR$(77)
                CALL MoRi
            END SELECT
        END SELECT
      LOOP
    LOOP UNTIL MTedit$ = CHR$(27)
    CALL PrTra(MtR%,DaC%,ColArr%(12),ColArr%(15))
8050    CALL ClEdSc
    SELECT CASE DatBck$
      CASE "E"
        LOCATE 3,20,0 : PRINT USING "Highway ID  : &"; LEFT$(HighwayNo$,10);
        LOCATE 4,20,0 : PRINT USING "Equation ID : &"; LEFT$(SubEqID$,10);
        LOCATE 5,20,0 : PRINT USING "Variable    : &"; LEFT$(Neck$,25)
        LOCATE 6,20,0 : PRINT USING "No. Variable: #"; NoVar%
      CASE "T"
        COLOR ColArr%(12),ColArr%(15)
        LOCATE 3,30,0   :  PRINT " PREDICTION VALUES "
        LOCATE 5,Col%,0 : PRINT USING " Highway ID: & "; Identification$(8)
    END SELECT
    COLOR ColArr%(12),ColArr%(15)
    LOCATE 24,16,0  :  PRINT " F1 : Help ";
    LOCATE 24,28,0  :  PRINT " F4 : Edit ";
    LOCATE 24,40,0  :  PRINT " ESC : Quit ";
    LOCATE 24,53,0  :  PRINT " F9 : End ";
8065 END IF
END SUB
'.............................................................................
SUB PrTra(MtR%,DaC%,Fore%,Back%)
8070  SHARED MtC%,Dat!(),NoData%,DaR%
      LOCATE MtR%,MtC%,0    ' go to the uppermost left data box
      COLOR Fore%,Back%
      IF MtR% = 8 THEN
        PRINT USING " ## "; NoData%
      ELSEIF DaC% = 1 THEN
        PRINT USING "  ####          "; Dat!(DaR%,DaC%)
      ELSEIF DaC% = 2 THEN
        PRINT USING " ###########    "; Dat!(DaR%,DaC%)
      END IF
END SUB
'.............................................................................
'the MoDo subroutine moves the cursor up or down for editing
SUB MoDo
8100  SHARED ColArr%(),MtR%,MtC%,DaR%,DaC%,MtLaR%
      CALL PrTra(MtR%,DaC%,ColArr%(12),ColArr%(15))
      IF MtR% = 8 THEN
        MtR% = 12  :  MtC% = 25
      ELSEIF DaC% = 1 THEN
        INCR MtR%  :  INCR DaR%
      ELSEIF DaC% = 2 THEN
        INCR MtR%  :  INCR DaR%
      END IF
```

```
      IF MtR% > MtLaR% THEN  ' this loop moves the cursor to the
        MtR% = 12              ' uppermost data box if the box where the
        DaR% = 1              ' cursor was located when this routine was
      END IF                   ' called, is reached
      IF MtR% = MtLaR% AND DaC% = 2 THEN
        MtR% = 12
        DaR% = 1
      END IF
      CALL PrTra(MtR%,DaC%,ColArr%(10),ColArr%(12))
END SUB
'.........................................................................
'the MoRi subroutine moves the cursor left or right
SUB MoRi
8640  SHARED ColArr%(),MtR%,MtC%,DaR%,DaC%,MtLaR%
      IF MtR% <> 8 THEN
        CALL PrTra(MtR%,DaC%,ColArr%(12),ColArr%(15))
        INCR DaC%  :  MtC% = MtC% + 19
        IF DaC% > 2 THEN    ' if on the ESAL column, then go to the
          DaC% = 1  :  MtC% = 25         ' Year column
        END IF
        IF MtR% = MtLaR% AND DaC% = 2 THEN
          DaC% = 1  :  MtC% = 25
        END IF
        CALL PrTra(MtR%,DaC%,ColArr%(10),ColArr%(12))
      END IF
END SUB
'-------------------------------------------------------------------------
'the following subroutine is used to erase or change any data on the
'PredictedTraffic subroutine
SUB PreTrEd
8680 SHARED ColArr%(),Help$,RowPo%,NScr%,NxScr%,lastrow%,LastVehicles%
     SHARED ColPo%,ColIN%,RowEd%,ColEd%,InEd%,ESAL#(),pt1%,pt8%,ChT$
     CALL EditScreen
     COLOR ColArr%(15),ColArr%(1)
     LOCATE 24,11,0
     PRINT "Switch to MANUAL TRAFFIC input (y/n) ?";
     COLOR ColArr%(12),ColArr%(15)
     LOCATE 24,52,0  : PRINT "   ";
     DO
8685   LOCATE 24,53,1
       WHILE NOT INSTAT  :  WEND
       Switch$ = INKEY$
       SELECT CASE Switch$
         CASE "Y","y"
           Help$ = "Q"
           ChT$ = "Y"
           EXIT LOOP
         CASE "N","n"
           COLOR ColArr%(15),ColArr%(1)
           LOCATE 24,11,0  :  PRINT SPACE$(45);
           EXIT LOOP
       END SELECT
```

```
     LOOP
     IF Help$ <> "Q" THEN
       Help$ = "E"    ' to reposition the cursor when going back
'depending on which screen the user was before calling the change subroutine
'the controlling variables are defined in the following IF block
       RowPo% = 11
       IF NScr% = NxScr% THEN
         lastrow% = LastVehicles% + 10
       ELSE
         lastrow% = 20
       END IF
       ColPo% = (Nscr% * 10) + 1
       ColIN% = ColPo%
       RowEd% = RowPo%
       ColEd% = 41
       InEd% = 1
       DO
8690     CALL PrDat(InEd%,ColArr%(15),ColArr%(12))
         DO
8700       WHILE NOT INSTAT : WEND
           Change$ = INKEY$
           SELECT CASE Change$
             CASE CHR$(13)
               COLOR ColArr%(12),ColArr%(15)
               LOCATE RowEd%,ColEd%,0  :  PRINT SPACE$(10)
               CALL EditDouble(EdValue#,RowEd%,(ColEd%+1),13)
               ESAL#(ColIN%,InEd%) = EdValue#
               EXIT LOOP
             CASE CHR$(27)
               EXIT LOOP
             CASE ELSE
               SELECT CASE MID$(Change$,2,1)
                 CASE CHR$(72),CHR$(80)
                   CALL DownUp
                 CASE CHR$(75),CHR$(77)
                   CALL RightLeft
               END SELECT
           END SELECT
         LOOP
         IF Change$ = CHR$(27) THEN EXIT LOOP
       LOOP
'restore the input screen
       CALL PrDat(InEd%,ColArr%(12),ColArr%(15))
       CALL ClEdSc
       COLOR ColArr%(12),ColArr%(15)
       LOCATE 2,32,0  :  PRINT "  TRAFFIC DATA  "
       LOCATE 3,32,0  :  PRINT " AASHTO  Method "
       LOCATE 4,15,0
       PRINT " Worksheet for calculating 18-kip ESAL applications "
       LOCATE 24,16,0  :  PRINT " F1 : Help ";
       LOCATE 24,28,0  :  PRINT " F6 : Edit ";
       LOCATE 24,40,0  :  PRINT " ESC : Quit ";
```

```
      LOCATE 24,53,0  :  PRINT " F9 : End ";
8800 END IF
END SUB
'..........................................................................
'print the values on the cell
SUB PrDat(InEd%,Fore%,Back%)
8850 SHARED RowEd%,ColEd%,ColIN%,ESAL#()
     COLOR Fore%,Back%
     LOCATE RowEd%,ColEd%,0  :  PRINT STRING$(10,0)
     LOCATE RowEd%,ColEd%,0
     SELECT CASE InEd%
       CASE 3
         PRINT USING "##.###### "; ESAL#(ColIN%,InEd%)
       CASE 2
         PRINT USING " ##.##     "; ESAL#(ColIN%,InEd%)
       CASE 1
         PRINT USING " ######.# " ;ESAL#(ColIN%,InEd%)
     END SELECT
END SUB
'..........................................................................
'the DownUp subroutine moves the cursor down or up
SUB DownUp
9100  SHARED RowEd%,lastrow%,pt1%,RowPo%,ColPo%,ColIN%,InEd%,pt8%,ColArr%()
      SELECT CASE RowEd%
        CASE lastrow%,pt1%
          CALL PrDat(InEd%,ColArr%(12),ColArr%(15))
          RowEd% = RowPo%  :  ColIN% = ColPo%
        CASE ELSE
'change the color of the data box along the movement of the cursor
          CALL PrDat(InEd%,ColArr%(12),ColArr%(15))
          INCR RowEd%  :  INCR ColIN%
'if the cursor gets to the cursor position before the change routine
'was called, then go to the uppermost value in the column
          IF RowEd% = pt1% AND InEd% >= pt8% THEN
            RowEd% = RowPo%  :  ColIN% = ColPo%
          END IF
      END SELECT
      CALL PrDat(InEd%,ColArr%(15),ColArr%(12))
END SUB
'..........................................................................
'the RightLeft subroutine moves the cursor left or right
SUB RightLeft
9400  SHARED RowEd%,lastrow%,pt1%,RowPo%,ColPo%,ColIN%,InEd%,pt8%,ColArr%()
      SHARED ColEd%
'to move from the ESAL factor column to the current traffic column
      IF InEd% = 3 THEN
        CALL PrDat(InEd%,ColArr%(12),ColArr%(15))
        ColEd% = 41  :  InEd% = 1
      ELSE
'change the color of the current column and move the next one
        CALL PrDat(InEd%,ColArr%(12),ColArr%(15))
        INCR ColEd%,11
```

```
        INCR InEd%
      END IF
      IF RowEd% = pt1% THEN
        IF InEd% = pt8% THEN
          InEd% = 1
          ColEd% = 41
        END IF
      END IF
      CALL PrDat(InEd%,ColArr%(15),ColArr%(12))
END SUB
'-------------------------------------------------------------------------------
'the following subroutine is used to modify values on Structural Data routine
SUB StrDatEd
9905  SHARED Help$,SN$,InCase%,EditSD%,choice%,ResilientModulus#,CBRvalue#
      SHARED Rvalue#,Relity!,StandDev!,Reliability$,ColArr%(),SNrow%,SNcol%
      SHARED SNcoeff%,SNchoice%,LayerCoef!(),Layer%,LayerIN%
      Help$ = "S"
      IF SN$ <> "Y" THEN
        CALL EditScreen
        InCase% = 1
        CALL SDown     ' start by positioning the cursor in the Mr choices
        DO
9910      WHILE NOT INSTAT  :  WEND  ' wait for a cursor key to be pressed
          SDchange$ = INKEY$  ' by pressing RETURN changes are made
          SELECT CASE SDchange$
            CASE CHR$(13)
'the following SELECT block asks the user to enter the new value
              DO
                COLOR ColArr%(15),ColArr%(12)
9960            SELECT CASE EditSD%
                  CASE 1   ' to change the Mr choice
                    COLOR ColArr%(15),ColArr%(1)
                    FOR Cl% = 10 TO 12      ' delete the old Mr choices
                      LOCATE Cl%,48,0
                      PRINT SPACE$(25)
                    NEXT Cl%
                    COLOR ColArr%(15),ColArr%(12)
                    DO
9970                  LOCATE 9,48,0  :  PRINT "   "
                      CALL EditInteger(EdValue%,9,49,1)
                      choice% = EdValue%
'check that a correct choice% value was entered
                      SELECT CASE choice%
                        CASE 1,2,3
                          EditSD% = 2
                          COLOR ColArr%(12),ColArr%(15)
                          LOCATE 9,48,0  :  PRINT USING " # "; choice%
                          EXIT LOOP
                      END SELECT
                    LOOP
                  CASE 2   ' to enter a new value for Mr, CBR, or R-value
                    SELECT CASE choice%   ' SELECT block # 2
```

```
                CASE 1    ' enter a new value for Mr
                  DO
9985                LOCATE 10,48,0  :  PRINT SPACE$(10)
                    CALL EditDouble(EdValue#,10,49,8)
                    ResilientModulus# = EdValue#
                  LOOP UNTIL ResilientModulus# <> 0
                CASE 2    ' enter a new value for CBR
                  DO
9990                LOCATE 11,48,0  :  PRINT "      "
                    CALL EditDouble(EdValue#,11,49,3)
                    CBRvalue# = EdValue#
                  LOOP UNTIL CBRvalue# <> 0
                  COLOR ColArr%(12),ColArr%(15)
                  ResilientModulus# = 1500 * CBRvalue#
                  LOCATE 11,57,0    ' print the new value for Mr
                  PRINT USING " Mr = ########"; ResilientModulus#
                CASE 3    ' enter a new value for R-value
                  DO
9995                LOCATE 12,48,0  :  PRINT "      "
                    CALL EditDouble(EdValue#,12,49,8)
                    Rvalue# = EdValue#
                  LOOP UNTIL Rvalue# <> 0
                  COLOR ColArr%(12),ColArr%(15)
                  ResilientModulus# = 1000 + (555 * Rvalue#)
                  LOCATE 12,57,0  ' print the new Mr
                  PRINT USING " Mr = ########"; ResilientModulus#
              END SELECT    ' end of SELECT block # 2
              CALL SDown
              EXIT LOOP
            CASE 3      ' to change the reliability value
              LOCATE 14,48,0  :  PRINT "      "
              CALL EditReal(EdValue!,14,49,5)
              Relity! = EdValue!
              LOCATE 14,48,0  :  PRINT USING " ##.## "; Relity!
              CALL Reliability(Relity!,StandDev!)
              COLOR ColArr%(12),ColArr%(15)
              LOCATE 14,57,0  :  PRINT USING " Zr =  +#.### "; StandDev!
              EXIT LOOP
          END SELECT    ' end of SELECT block 1
        LOOP
     CASE CHR$(27)  'go back to enter more data
        COLOR ColArr%(12),ColArr%(15)
        SELECT CASE EditSD%
          CASE 1
            LOCATE 9,48,0  :  PRINT USING " # "; choice%
          CASE 2
            CALL ColorMR(choice%,ColArr%(12),ColArr%(15))
          CASE 3
            LOCATE 14,48,0  :  PRINT USING " ##.## "; Relity!
        END SELECT
        EXIT LOOP
      CASE ELSE
```

```
            SELECT CASE MID$(SDchange$,2,1)
              CASE CHR$(72),CHR$(80)
                CALL SDown
            END SELECT
        END SELECT
      LOOP
'for changing data on the structural number routine
    ELSE
10000   CALL EditScreen
        COLOR ColArr%(15),ColArr%(12)
        SNrow% = 13 : SNcol% = 30 : SNcoeff% = 17   'controlling variables
        LOCATE SNrow%,37,0   ' put the cursor on the SNchoice box
        PRINT USING " # "; SNchoice%
        DO
10020     DO  :  LOOP UNTIL INSTAT
          SNchange$ = INKEY$
          SELECT CASE SNchange$
            CASE CHR$(13)     ' pressing RETURN enables the user to change data
              COLOR ColArr%(15),ColArr%(12)
              SELECT CASE SNrow%  'depending on the position, change a value
                CASE 13   'change the structural number selection
                  DO
10025               LOCATE 13,37,0  :  PRINT "   "
                    CALL EditInteger(EdValue%,13,38,1)
                    SNchoice% = EdValue%
                    SELECT CASE SNchoice%
                      CASE 1,2  'for a new value of SN selection
                        COLOR ColArr%(4),ColArr%(1)
                        FOR Cl% = 16 TO 19
                          LOCATE Cl%,11,0
                          PRINT SPACE$(57)
                        NEXT Cl%
                        Help$ = "C"
                    END SELECT
                  LOOP UNTIL Help$ = "C"
                CASE ELSE  'change any other number on the SNcoeff routine
                  LOCATE SNrow%,SNcol%,0
                  PRINT SPACE$(10)
                  LOCATE SNrow%,(SNcol%+2),1
                  CALL EditReal(EdValue!,SNrow%,(SNcol%+2),6)
                  LayerCoef!(SNrow%,SNcoeff%) = EdValue!
              END SELECT
              IF Help$ <> "C" THEN LOCATE SNrow%,(SNcol%+2),1
            CASE CHR$(27)
              COLOR ColArr%(12),ColArr%(15)
              SELECT CASE SNrow%
                CASE 13
                  LOCATE 13,37,0  :  PRINT USING " # "; SNchoice%
                CASE ELSE
                  LOCATE SNrow%,SNcol%,0
                  PRINT USING " ##.###   "; LayerCoef!(SNrow%,SNcoeff%)
              END SELECT
```

```
            EXIT LOOP
          CASE ELSE
            SELECT CASE MID$(SNchange$,2,1)
              CASE CHR$(72),CHR$(80)
                CALL SNdown
              CASE CHR$(75),CHR$(77)
                CALL SNright
            END SELECT
        END SELECT
      LOOP UNTIL Help$ = "C"
    END IF
10100 CALL ClEdSc
      COLOR ColArr%(12),ColArr%(15)    ' restore the upper part of the main screen
      LOCATE 3,23,0   :  PRINT " PREDICTED SERVICEABILITY MODULE "
      LOCATE  4,34,0  :  PRINT " DATA BASE "
      LOCATE 24,16,0  :  PRINT " F1 : Help ";
      LOCATE 24,28,0  :  PRINT " F8 : Edit ";
      LOCATE 24,40,0  :  PRINT " ESC : Quit ";
      LOCATE 24,53,0  :  PRINT " F9 : End ";
END SUB
'...............................................................................
'break the source code in segments of 64K
      $SEGMENT
'...............................................................................
'the following routine moves the cursor vertically for the structural data
SUB SDown
10200 SHARED InCase%,EditSD%,Relity!,choice%,Reliability$,ColArr%()
      SELECT CASE InCase%
        CASE 1  'modify the selection for a resilient modulus input
          IF EditSD% = 3 THEN
            COLOR ColArr%(12),ColArr%(15)
            LOCATE 14,48,0
            PRINT USING " ##.## "; Relity!
          END IF
          COLOR ColArr%(15),ColArr%(12)
          LOCATE 9,48,0  :  PRINT USING " # "; choice%
          INCR InCase%   :  EditSD% = 1
        CASE 2   'change a value of Mr
          COLOR ColArr%(12),ColArr%(15)
          LOCATE 9,48,0  :  PRINT USING " # "; choice%
          CALL ColorMR(choice%,ColArr%(15),ColArr%(12))
          INCR InCase%  :  EditSD% = 2
10250   CASE 3  'edit a value of reliability
          CALL ColorMR(choice%,ColArr%(12),ColArr%(15))
          COLOR ColArr%(15),ColArr%(12)
          SELECT CASE Reliability$
            CASE "Y"
              LOCATE 14,48,0
              PRINT USING " ##.## "; Relity!
              EditSD% = 3  :  InCase% = 1
            CASE ELSE
              LOCATE 9,48,0
```

```
            PRINT USING " # "; choice%
            EditSD% = 1  :  InCase% = 2
          END SELECT
        END SELECT
END SUB
'.........................................................................
'this subroutine changes the color for the Mr choices
SUB ColorMR(choice%,Fore%,Back%)
10275 SHARED ResilientModulus#,CBRvalue#,Rvalue#
      COLOR Fore%,Back%
      LOCATE (choice%+9),48,0
      SELECT CASE choice%
        CASE 1
          PRINT USING " #######  "; ResilientModulus#
        CASE 2
          PRINT USING " ### "; CBRvalue#
        CASE 3
          PRINT USING " ###  "; Rvalue#
      END SELECT
END SUB
'.........................................................................
'the following subroutine moves the cursor vertically when on the structural
'number subroutine
SUB SNdown
10300 SHARED SNchoice%,SNrow%,SNcol%,LayerCoef!(),SNcoeff%,Layer%,LayerIN%
      SHARED ColArr%()
      IF SNchoice% = 1 THEN
        IF SNrow% = 13 THEN       ' if on the SNchoice box
          LOCATE SNrow%,37,0
          COLOR ColArr%(12),ColArr%(15)
          PRINT USING " # "; SNchoice%
          SNrow% = 17
          COLOR ColArr%(15),ColArr%(12)
          LOCATE SNrow%,SNcol%,0
          PRINT USING " ##.###   "; LayerCoef!(SNrow%,SNcoeff%)
        ELSE
          COLOR ColArr%(12),ColArr%(15)
          LOCATE SNrow%,SNcol%,0
          PRINT USING " ##.###   "; LayerCoef!(SNrow%,SNcoeff%)
          INCR SNrow%
10350     SELECT CASE SNcoeff%
            CASE 19
              IF SNrow% = 17 THEN
                SNrow% = 18
              ELSEIF SNrow% = Layer% THEN
                DECR SNrow%
              END IF
            CASE LayerIN%
              IF SNrow% >= Layer% THEN SNrow% = 17
            CASE ELSE
              IF SNrow% > 19 THEN SNrow% = 17
              IF SNrow% > Layer% THEN SNrow% = 17
```

```
            IF SNrow% = Layer% AND SNcoeff% > LayerIN% THEN SNrow% = 17
          END SELECT
          COLOR ColArr%(15),ColArr%(12)
          LOCATE SNrow%,SNcol%,0  ' locate and highlight the lower box
          PRINT USING " ##.###   "; LayerCoef!(SNrow%,SNcoeff%)
        END IF
      END IF
END SUB
'.........................................................................
'the following subroutine moves the cursor horizontally when on the structural
'number subroutine
SUB SNright
10400 SHARED SNchoice%,SNrow%,SNcol%,LayerCoef!(),SNcoeff%,Layer%,LayerIN%
      SHARED ColArr%()
      IF SNrow% <> 13 THEN
        COLOR ColArr%(12),ColArr%(15)
        LOCATE SNrow%,SNcol%,0
        PRINT USING " ##.###   " ;LayerCoef!(SNrow%,SNcoeff%)
        INCR SNcol%,11
        INCR SNcoeff%
        IF SNcoeff% > 19 THEN
          SNcoeff% = 17
          SNcol% = 30
        END IF
10450   IF SNrow% = Layer% AND SNcoeff% = LayerIN% THEN
          SNcoeff% = 17
          SNcol% = 30
        END IF
        IF SNrow% > Layer% AND SNcoeff% <= LayerIN% THEN
          SNcoeff% = 17
          SNcol% = 30
        END IF
        IF SNrow% = 17 AND SNcoeff% = 19 THEN
          SNcoeff% = 17
          SNcol% = 30
        END IF
        COLOR ColArr%(15),ColArr%(12)
        LOCATE SNrow%,SNcol%,0
        PRINT USING " ##.###   "; LayerCoef!(SNrow%,SNcoeff%)
      END IF
END SUB
'-------------------------------------------------------------------------
'the following subroutine is used to change data on the DeflectionData routine
SUB DefEdit
12000 SHARED Help$,DeflectionChange$,D%,LastRow%,Drow%,Deflection!(),Dscr$
      SHARED PerfChoice%,ColArr%()
      DeflectionChange$ = "yes"
      CALL EditScreen  ' display the edit commands
      DeflectionChange$ = ""
      LastRow% = D%  :  Drow% = 17  : Help$ = "E"
      DO
        CALL PrnDef(Drow%,ColArr%(15),ColArr%(12))
```

```
        DO
          WHILE NOT INSTAT  :  WEND
          Dchange$ = INKEY$
          SELECT CASE Dchange$
            CASE CHR$(13)
              LOCATE Drow%,50,0  :  PRINT SPACE$(10)
              CALL EditReal(EdValue!,Drow%,52,8)
              Deflection!(Drow%) = EdValue!
              EXIT LOOP
            CASE CHR$(27)   ' press ESC to return to data entry
              Help$ = "Q"
              EXIT LOOP
12150       CASE ELSE
              SELECT CASE MID$(Dchange$,2,1)
                CASE CHR$(72),CHR$(80)
                  CALL DefDown
              END SELECT
          END SELECT
        LOOP
        IF Dchange$ = CHR$(27) THEN EXIT LOOP
      LOOP
      CALL PrnDef(Drow%,ColArr%(12),ColArr%(15))
      Dscr$ = "Y"   ' restore the screen
      COLOR ColArr%(15),ColArr%(1)
      CALL PerfScr
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 8,38,0   :  PRINT USING "  #  "; PerfChoice%
      LOCATE 24,16,0  :  PRINT " F1 : Help ";
      LOCATE 24,28,0  :  PRINT " F8 : Edit ";
      LOCATE 24,40,0  :  PRINT " ESC : Quit ";
12200 LOCATE 24,53,0  :  PRINT " F9 : End ";
END SUB
'.........................................................................
'the following routine moves the cursor up and down
SUB DefDown
30550 SHARED ColArr%(),Drow%,LastRow%
      CALL PrnDef(Drow%,ColArr%(12),ColArr%(15))
      INCR Drow%
      IF Drow% >= LastRow% THEN Drow% = 17
      CALL PrnDef(Drow%,ColArr%(15),ColArr%(12))
END SUB
'.........................................................................
SUB PrnDef(Drow%,Fore%,Back%)
30500 SHARED Deflection!()
      COLOR Fore%,Back%   ' to place the cursor in the data box
      LOCATE Drow%,50,0
      IF Drow% = 17 THEN
        PRINT USING " ######## "; Deflection!(Drow%)
      ELSE
        PRINT USING " ##.##### "; Deflection!(Drow%)
      END IF
END SUB
```

```
'-------------------------------------------------------------------------------
'to make changes during NewEquation routine
SUB NeEqE
30450 SHARED Operation!(),NoVar%,Operation$()
      Operation$(10,NoVar%) = ""
      FOR Init1% = 1 TO 9
        Operation!(Init1%,NoVar%) = 1000000
      NEXT Init1%
END SUB
'-------------------------------------------------------------------------------
SUB ClEdSc
30400 SHARED ColArr%()
      COLOR ColArr%(15),ColArr%(1)
      FOR Cl% = 2 TO 4
        LOCATE Cl%,10,0
        PRINT SPACE$(66)
      NEXT Cl%
END SUB
'-------------------------------------------------------------------------------
Graph:
'determine the type of graphics card to use
      ON ERROR GOTO ErrorTrap
9775  SCREEN 2  'Hercules or CGA in black & white
      DEF SEG = 0
      VideoMode% = PEEK(&H449)
      VideoGraph% = VideoMode%
9780  SCREEN 8  'CGA with color
      VideoGraph% = 14
9785  SCREEN 9  'EGA
      VideoGraph% = 16
9790  ON ERROR GOTO 0
      RETURN
'-------------------------------------------------------------------------------
'determine the type of monitor being used: monochrome or color
SUB Monitor(VideoMode%)
30300 SHARED ColArr%()
      DEF SEG = 0
      VideoMode% = PEEK(&H449)
      DIM DYNAMIC ColArr%(1:20)
      SELECT CASE VideoMode%
        CASE 5,6,7,13
          ColArr%(1) = 0   : ColArr%(4) = 7   : ColArr%(5) = 16
          ColArr%(6) = 15  : ColArr%(10) = 7  : ColArr%(12) = 0
          ColArr%(13) = 7  : ColArr%(14) = 15 : ColArr%(15) = 15
          ColArr%(16) = 16 : ColArr%(17) = 7  : ColArr%(18) = 23
          ColArr%(19) = 0  : ColArr%(20) = 7
        CASE ELSE
          ColArr%(1) = 1   : ColArr%(4) = 4   : ColArr%(5) = 31
          ColArr%(6) = 10  : ColArr%(10) = 10 : ColArr%(12) = 4
          ColArr%(13) = 4  : ColArr%(14) = 14 : ColArr%(15) = 15
          ColArr%(16) = 20 : ColArr%(17) = 26 : ColArr%(18) = 28
          ColArr%(19) = 30 : ColArr%(20) = 31
```

```
     END SELECT
END SUB
'-----------------------------------------------------------------------
'routine used to enter an integer value
SUB EditInteger(EdValue%,EdRow%,EdCol%,EdNoChr%)
30000 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue% = CINT(VAL(EdStr$))
END SUB
'-----------------------------------------------------------------------
'routine used to enter a long integer value
SUB EditLong(EdValue&,EdRow%,EdCol%,EdNoChr%)
30050 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue& = CLNG(VAL(EdStr$))
END SUB
'-----------------------------------------------------------------------
'routine used to enter a real value
SUB EditReal(EdValue!,EdRow%,EdCol%,EdNoChr%)
30100 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue! = CSNG(VAL(EdStr$))
END SUB
'-----------------------------------------------------------------------
'routine used to enter a string
SUB EditDouble(EdValue#,EdRow%,EdCol%,EdNoChr%)
30150 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
      EdValue# = CDBL(VAL(EdStr$))
END SUB
'-----------------------------------------------------------------------
'routine used to enter data
SUB EditString(EdStr$,EdRow%,EdCol%,EdNoChr%)
30200 CALL EditData(EdRow%,EdCol%,EdStr$,EdNoChr%)
END SUB
'-----------------------------------------------------------------------
'this routine enters a string of up to MaxNoChr% characters
SUB EditData(EdRow%,EdCol%,EdInStr$,EdNoChr%)
30250 LOCAL LetPos%,NoChr%,Lines%,StartCol%
      EdInStr$ = STRING$(EdNoChr%,0) : LetPos% = 1
      StartCol% = EdCol%
      NoChr% = 1  :  EdCol% = StartCol%
      DO WHILE NoChr% <= EdNoChr%
        DO
          LOCATE EdRow%,EdCol%,1
        LOOP UNTIL INSTAT
        Letter$ = INKEY$
        SELECT CASE Letter$
          CASE CHR$(27)
            EXIT SUB
          CASE CHR$(13)
            EXIT LOOP
          CASE CHR$(8)
            IF EdCol% > StartCol% THEN   'avoid backspacing
              DECR EdCol% : DECR NoChr%  :  DECR LetPos%
              LOCATE EdRow%,EdCol%,0  :  PRINT SPACE$(1)
```

```
            MID$(EdInStr$,LetPos%,1) = CHR$(0)
          END IF
        CASE ELSE
          IF LEN(Letter$) = 1 THEN
            LOCATE EdRow%,EdCol%,0
            PRINT Letter$
            MID$(EdInStr$,LetPos%,1) = Letter$
            INCR EdCol% : INCR NoChr%  :  INCR LetPos%
          END IF
      END SELECT
    LOOP
    EdInStr$ = LEFT$(EdInStr$,LetPos%-1)
END SUB
'................................................................................
```

Appendix A.10
PBAHELP.BAS

```
'-----------------------------------------------------------------------
'
'                              PBAHELP.BAS
'
' the following program was coded using Turbo BASIC version 1.1
' this file contains the subroutines used during the execution of the
' main file "PBAASHTO.BAS" to obtain help, and to create screens.
'-----------------------------------------------------------------------
'turn key trapping off
     $EVENT OFF
'-----------------------------------------------------------------------
'this routine displays a procedure-sensitive help screen
SUB PBAhelp(OnHelp$)
13000 SELECT CASE OnHelp$
        CASE "MENU"
          HelpFile$ = "CHAP2.DOC"
          Title$ = "Main Menu"
        CASE "IData"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "Identification Data"
        CASE "AASHTO"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "AASHTO Design Equations"
        CASE "DEFLECTION"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "AASHTO Deflection Equation"
        CASE "PBAindex"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "PBA Index"
        CASE "DefIndex"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "PBA Index"
        CASE "PREDICT"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "Traffic Prediction"
        CASE "CAR"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "Vehicle Types"
        CASE "USER"
          HelpFile$ = "CHAP5.DOC"
          Title$ = "User's Performance Equation"
        CASE "Xvariable"
          HelpFile$ = "CHAP3.DOC"
          Title$ = "Performance Independent Variable"
      END SELECT
      CALL HelpScr(HelpFile$,Title$)
END SUB
'-----------------------------------------------------------------------
'this routine opens an ASCII file and displays its content
SUB HelpScr(HelpFile$,Title$)
      shared wrow%(),wrows%(),wcol%(),wcols%(),wattr%(),wbrdr%(),wshdw%()
      shared scrn%(),wptr(),Help$,InHelp$,ColArr%(),FileErr$
```

```
      LOCAL ViewFile$()
      InHelp$ = "Y" : FileErr$ = HelpFile$
4000  LOCATE 19,26,0
4050  OPEN HelpFile$ FOR INPUT AS # 5
      IF Help$ <> "H" THEN
        ColScr% = FNATTR%(ColArr%(5),ColArr%(6))
        CALL MsgWindow(3,5,20,70,ColScr%,2,1,0)
        Title$ = " " + Title$ + " "
        Guide$ = " ENTER : next   ESC : quit "
        CALL TITLEWINDOW(5,Guide$)
        CALL TITLEWINDOW(2,Title$)
        GoBack$ = CHR$(13)  'control variable
4200    DO   'do until the user press ESC to exit, or the file is read totally
          Row% = 0
          CALL ClearWindow
          DIM DYNAMIC ViewFile$(1:17)
          DO
            DO
              SELECT CASE GoBack$
                CASE CHR$(13)
                  INCR Row%
4250              LINE INPUT # 5, ViewFile$(Row%)
                  CALL PrtWindow(Row%,2,ViewFile$(Row%))
              END SELECT
              IF Row% >= 17 OR EOF(5) THEN EXIT LOOP
            LOOP
            DO : LOOP UNTIL INSTAT
            GoBack$ = INKEY$
            IF GoBack$ = CHR$(27) THEN EXIT LOOP
          LOOP UNTIL GoBack$ = CHR$(13)
          ERASE ViewFile$
          IF GoBack$ = CHR$(27) THEN EXIT LOOP
        LOOP UNTIL EOF(5)
4275    CLOSE # 5
        CALL REMOVEWINDOW
        erase wrow%,wrows%,wcol%,wcols%,wattr%,wbrdr%,wshdw%,scrn%,wptr
      END IF
4300  Help$ = "Y"
END SUB
'-----------------------------------------------------------------------------
'display the error message screen
SUB MsgWindow(ROW%,COL%,ROWS%,COLS%,ATTR%,BRDRSEL%,SHADOW%,ZOOM%)
12950 SHARED wrow%(),wrows%(),wcol%(),wcols%(),wattr%(),InHelp$
      SHARED wbrdr%(),wshdw%(),scrn%(),wptr(),ColArr%(),ErMsg$()
      mw% = 3  :  ScrnArray = 8000
      dim dynamic wrow%(mw%),wrows%(mw%),wcol%(mw%),wcols%(mw%),wattr%(mw%)
      dim dynamic wbrdr%(mw%),wshdw%(mw%),scrn%(ScrnArray),wptr(mw%)
      CALL MAKEWINDOW(ROW%,COL%,ROWS%,COLS%,ATTR%,BRDRSEL%,SHADOW%,ZOOM%)
      IF InHelp$ <> "Y" THEN
        CALL PRTCWINDOW(1,ErMsg$(3))
        CALL PRTCWINDOW(2,ErMsg$(1))
        CALL PRTCWINDOW(3,ErMsg$(2))
```

```
        CALL PRTCWINDOW(4,ErMsg$(4))
        DO
          ErMsg$(5) = INKEY$
        LOOP UNTIL ErMsg$(5) = CHR$(27)
        CALL REMOVEWINDOW
        erase wrow%,wrows%,wcol%,wcols%,wattr%,wbrdr%,wshdw%,scrn%,wptr
      END IF
      InHelp$ = ""
END SUB
'--------------------------------------------------------------------------------
'the following subroutine creates the screen for inputing the structural data
SUB StrDatScr
30600 SHARED ColArr%(),Identification$(),SN$
      CALL Border("Y")
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 24,16,0 :  PRINT " F1 : Help ";
      LOCATE 24,28,0 :  PRINT " F8 : Edit ";
      LOCATE 24,40,0 :  PRINT " ESC : Quit ";
      LOCATE 24,53,0 :  PRINT " F9 : End ";
      Col% = LEN(Identification$(8)) + 15
      Col% = (80 - Col%) / 2
      LOCATE 6,Col%,0 : PRINT USING " Highway ID : & "; Identification$(8);
      IF SN$ = "Y" THEN
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 8,10,0 : PRINT "Structural Number :"
        LOCATE 9,12,0
        PRINT "1. Estimated based on structural layer coefficients (ai),"
        LOCATE 10,15,0
        PRINT "layer thickness (Di), and drainage coefficients (mi)"
        LOCATE 11,12,0
        PRINT "2. Value from Design Specifications"
        LOCATE 13,17,0 : PRINT "Select a choice : "
        COLOR ColArr%(4),ColArr%(1)
        LOCATE 15,10,0 : PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
        FOR b% = 16 TO 19
          LOCATE b%,10,0 : PRINT CHR$(179) + STRING$(58,0) + CHR$(179)
        NEXT b%
        LOCATE 20,10,0 : PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
      ELSE
        COLOR ColArr%(15),ColArr%(1)
        LOCATE 8,20,0  :  PRINT "Roadbed Resilient Modulus (Mr) : "
        LOCATE 9,22,0  :  PRINT "Select a choice :"
        LOCATE 10,25,0 :  PRINT "1. Mr value "
        LOCATE 11,25,0 :  PRINT "2. CBR correlation"
        LOCATE 12,25,0 :  PRINT "3. R-value correlation"
        LOCATE 14,20,0 :  PRINT "Reliability (R, percent)="
        LOCATE 15,25,0 :  PRINT "<default : 0.95>"
        LOCATE 17,20,0 :  PRINT "Standard Deviation (So) ="
        LOCATE 18,25,0 :  PRINT "<default : 0.45>"
      END IF
END SUB
'--------------------------------------------------------------------------------
```

```
'the following subroutine is used to create the screen for selecting a choice
'of Performance Measure
SUB PerfScr
30700 SHARED Dscr$,PerfEqn$,ColArr%()
      IF Dscr$ <> "Y" THEN
        COLOR ColArr%(15),ColArr%(1)
        CLS
      END IF
      LOCATE 2,10,0 : PRINT CHR$(218) + STRING$(58,196) + CHR$(191) + SPACE$(8)
      LOCATE 3,10,0 : PRINT CHR$(179) + STRING$(58,0) + CHR$(179) + SPACE$(8)
      LOCATE 4,10,0 : PRINT CHR$(195) + STRING$(58,196) + CHR$(180) + SPACE$(8)
      FOR Edge% = 5 TO 8
        LOCATE Edge%,10,0
        PRINT CHR$(179) + SPACE$(58) + CHR$(179) + "   "
      NEXT Edge%
      LOCATE 9,10,0  :  PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
      IF PerfEqn$ = "Y" THEN    'if called from performance
        LOCATE 9,10,0  : PRINT CHR$(179) + SPACE$(58) + CHR$(179) + "   "
        LOCATE 10,10,0 : PRINT CHR$(179) + SPACE$(58) + CHR$(179) + "   "
        LOCATE 11,10,0 : PRINT CHR$(192) + STRING$(58,196) + CHR$(217)
        EXIT SUB   'if called from Performance
      ELSE
        LOCATE 3,30,0 : PRINT " Performance Measure "
        LOCATE 5,20,0 : PRINT "1. Present Serviceability Index (PSI)"
        LOCATE 6,20,0 : PRINT "2. Deflection"
        LOCATE 7,20,0 : PRINT "3. Cancel data for current section"
        LOCATE 8,20,0 : PRINT "Enter a Choice : "
      END IF
END SUB
'-------------------------------------------------------------------------
' create the directions for the change screen
SUB EditScreen
30800 SHARED ColArr%(),DeflectionChange$,sdsayuda$,CarEd$,SN$
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 2,10,0  :  PRINT CHR$(218) + STRING$(64,196) + CHR$(191)
      LOCATE 3,10,0  :  PRINT CHR$(179) + SPACE$(64) + CHR$(179)
      LOCATE 4,10,0  :  PRINT CHR$(192) + STRING$(64,196) + CHR$(217)
      LOCATE 3,11,0  :  PRINT " " + CHR$(25) + " " + CHR$(24) + " : vertical "
      LOCATE 3,28,0  :  PRINT " " + CHR$(26) + " " + CHR$(27) + " : lateral "
      LOCATE 3,44,0
      COLOR ColArr%(16),ColArr%(10)  : PRINT " ENTER ";
      COLOR ColArr%(12),ColArr%(15) : PRINT ": edit "
      COLOR ColArr%(10),ColArr%(1)
      IF DeflectionChange$ = "yes" THEN
        LOCATE 24,11,0  :  PRINT SPACE$(58);
      ELSE
        LOCATE 24,6,0   :  PRINT SPACE$(69);
      END IF
      LOCATE 3,60,0 : PRINT " ESC : exit "
      COLOR ColArr%(12),ColArr%(15)
      IF sdsayuda$ = "yes" OR SN$ = "yes" THEN
        LOCATE 3,60,0 : PRINT SPACE$(12)
```

```
      ELSEIF CarEd$ = "Y" OR DeflectionChange$ = "yes" THEN
        LOCATE 3,28,0 : PRINT SPACE$(15)
      END IF
END SUB
'---------------------------------------------------------------------------
'create the data collection screen for the identification data
SUB Border(StrDat$)
30900 SHARED ColArr%()
      CALL Frame
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 3,24,0  :  PRINT " PREDICTED SERVICEABILITY MODULE "
      LOCATE  4,35,0 :  PRINT " DATA BASE "
      IF StrDat$ <> "Y" THEN
        LOCATE 24,17,0  :  PRINT " F1 : Help ";
        LOCATE 24,29,0  :  PRINT " F2 : Edit ";
        LOCATE 24,41,0  :  PRINT " F9 : End ";
        LOCATE 24,52,0  :  PRINT " ESC : Menu ";
        COLOR ColArr%(15),ColArr%(1)
        LOCATE  7,30,0  :  PRINT "Identification: "
        LOCATE  8,31,0  :  PRINT "Highway #       : "
        LOCATE  9,31,0  :  PRINT "District        : "
        LOCATE 10,31,0  :  PRINT "County          : "
        LOCATE 11,31,0  :  PRINT "Begin M.P.      : "
        LOCATE 12,31,0  :  PRINT "End M.P.        : "
        LOCATE 13,31,0  :  PRINT "Section EA      : "
        LOCATE 14,31,0  :  PRINT "EA/SJ           : "
        LOCATE 15,31,0  :  PRINT "Year            : "
        LOCATE 16,31,0  :  PRINT "Region          : "
        LOCATE 18,15,0  :  PRINT "Performance Measure:"
        LOCATE 19,15,0  :  PRINT "1. User's Performance, AASHTO-PSI"
        LOCATE 20,15,0  :  PRINT "2. AASHTO-Deflection"
      END IF
END SUB
'---------------------------------------------------------------------------
'screen for inputing the data for the PredictedTraffic subroutine
SUB ESALscr
31000 SHARED ColArr%(),Identification$()
      COLOR ColArr%(12),ColArr%(15)
      LOCATE 2,32,0  :  PRINT "  TRAFFIC DATA  "
      LOCATE 3,32,0  :  PRINT " AASHTO  Method "
      LOCATE 4,15,0
      PRINT " Worksheet for calculating 18-kip ESAL applications "
      Col% = LEN(Identification$(8)) + 15
      Col% = (80 - Col%) \ 2
      LOCATE 6,Col%,0 : PRINT USING " Highway ID : & "; Identification$(8);
      COLOR ColArr%(13),ColArr%(1)
      LOCATE 7,8,0
      PRINT CHR$(218) + STRING$(31,196) + CHR$(194) + STRING$(10,196)+CHR$(194)_
            + STRING$(10,196) + CHR$(194) + STRING$(10,196) + CHR$(191)
      EdTr$ = CHR$(179) + SPACE$(31) + CHR$(179) + SPACE$(10) + CHR$(179)_
              + SPACE$(10) + CHR$(179) + SPACE$(10) + CHR$(179)
      LOCATE 8,8,0  :  PRINT EdTr$
```

```
       LOCATE 9,8,0   :  PRINT EdTr$
       LOCATE 10,8,0
       PRINT CHR$(195) + STRING$(31,196) + CHR$(197) + STRING$(10,196)+CHR$(197)_
             + STRING$(10,196) + CHR$(197) + STRING$(10,196) + CHR$(180)
       FOR Ed% = 11 TO 20
         LOCATE Ed%,8,0
         PRINT EdTr$
       NEXT Ed%
       LOCATE 21,8,0
       PRINT CHR$(192)+STRING$(31,196) + CHR$(193) + STRING$(10,196) + CHR$(193)_
             +STRING$(10,196) + CHR$(193) + STRING$(10,196) + CHR$(217)
       COLOR ColArr%(15),ColArr%(1)
       LOCATE 8,17,0   :  PRINT " Vehicle Type "
       LOCATE 8,42,0   :  PRINT "current"
       LOCATE 8,54,0   :  PRINT "growth"
       LOCATE 8,66,0   :  PRINT "ESAL"
       LOCATE 9,42,0   :  PRINT "traffic"
       LOCATE 9,55,0   :  PRINT "rate"
       LOCATE 9,65,0   :  PRINT "factor"
END SUB
'-------------------------------------------------------------------------------
'the following routine displays the screen for inputting the deflection data
SUB DefScr
31100 SHARED ColArr%(),Identification$()
       COLOR ColArr%(15),ColArr%(1)
       LOCATE 12,10,0 : PRINT CHR$(218) + STRING$(58,196) + CHR$(191)
       LOCATE 13,10,0 : PRINT CHR$(179) + SPACE$(58) + CHR$(179)
       LOCATE 14,10,0 : PRINT CHR$(195) + STRING$(58,196) + CHR$(180)
       LOCATE 15,10,0 : PRINT CHR$(179) + SPACE$(58) + CHR$(179)
       LOCATE 16,10,0 : PRINT CHR$(195) + STRING$(58,196) + CHR$(180)
       FOR Edge% = 17 TO 22
         LOCATE Edge%,10,0
         PRINT CHR$(179) + SPACE$(58) + CHR$(179)
       NEXT Edge%
       LOCATE 23,10,0 : PRINT CHR$(195) + STRING$(58,196) + CHR$(180)
       LOCATE 24,10,0 : PRINT CHR$(179) + STRING$(58,0) + CHR$(179);
       LOCATE 25,10,0 : PRINT CHR$(192) + STRING$(58,196) + CHR$(217);
       COLOR ColArr%(12),ColArr%(15)
       LOCATE 13,32,0   :  PRINT " DEFLECTION DATA "
       Col% = LEN(Identification$(8)) + 15
       Col% = (80 - Col%) / 2
       LOCATE 15,Col%,0 : PRINT USING " Highway ID : & "; Identification$(8);
       LOCATE 24,16,0   :  PRINT " F1 : Help ";
       LOCATE 24,28,0   :  PRINT " F8 : Edit ";
       LOCATE 24,40,0   :  PRINT " ESC : Quit ";
       LOCATE 24,53,0   :  PRINT " F9 : End ";
END SUB
'-------------------------------------------------------------------------------
'These utilities are released into the public domain for anyone to use as they
'see fit; however, anyone who feels inspired and would like to contribute to
'my printer paper and computer supplies fund may feel free to do so.
'Rick Fothergill
```

```
'141 Oak Hill Road
'Pittsfield, MA 01201
'(413) 442-2456
'GEnie Address: R.FOTHERGILL
'Compuserve: 76210,443
'..........................................................................
sub MakeWindow(Row%,Col%,Rows%,Cols%,Attr%,BrdrSel%,Shadow%,Zoom%) static
  shared wrow%(),wrows%(),wcol%(),wcols%(),wattr%(),wbrdr%(),wshdw%(),scrn%()
  shared wptr(),LI,mw%,ScrnArray
  local r1%,r2%,c1%,c2%,colratio%,wsize
  select case shadow%
    Rem Left
    case = 1
      c1%=col%-2 : c2%=cols%+2 : r2%=rows%+1
    Rem Right
    case = 2
      c1%=col%   : c2%=cols%+2 : r2%=rows%+1
    case else
      c1%=col%   : c2%=cols%   : r2%=rows%
  end select
  wsize = (r2% * c2%) * 2
  LI = LI + 1
  Wptr(LI+1) = Wptr(LI)+WSize+1
  WRow%(LI)  = Row%
  WCol%(LI)  = Col%
  WRows%(LI) = Rows%
  WCols%(LI) = Cols%
  Wattr%(LI) = Attr%
  WBrdr%(LI) = BrdrSel%
  WShdw%(LI) = Shadow%
  Call Qsave(Row%,c1%,r2%,c2%,scrn%(Wptr(LI)))
  if zoom% = 1 then
    r1% = row% + (rows%\2)
    r2% = row% + rows%-(rows%\2)
    c1% = col% + (cols%\2)
    c2% = col% + cols%-(cols%\2)
    colratio% = (cols% \ rows%)+1
    if colratio% > 4 then colratio%=4
    do
      if r1%>row% then r1%=r1%-1
      if r2%<(row%+rows%) then r2%=r2%+1
      if c1%>col% then c1%=c1%-colratio%
      if c1%<col% then c1%=col%
      if c2%<(col%+cols%) then c2%=c2%+colratio%
      if c2%>(col%+cols%) then c2%=col%+cols%
      call Qbox(r1%,c1%,r2%-r1%,c2%-c1%,attr%,brdrsel%)
    loop until c1%=col% and c2%=col%+cols% and r1%=row% and r2%=row%+rows%
  else
    call Qbox(row%,col%,rows%,cols%,attr%,brdrsel%)
  end if
  select case shadow%
    rem Left
```

```
    case = 1
      call qfill(row%+1,col%-2,rows%-1,2,asc(" "),0)
      call qfill(row%+rows%,col%-2,1,cols%,asc(" "),0)
    rem Right
    case = 2
      call qfill(row%+1,col%+cols%,rows%-1,2,asc(" "),0)
      call qfill(row%+rows%,col%+2,1,cols%,asc(" "),0)
    case else
  end select
end sub

sub TitleWindow(dir%,title$) static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
  select case dir%
    rem UpperLeft
    case = 1
      call qprint(wrow%(LI),wcol%(LI)+2,title$,wattr%(LI))
    rem UpperCenter
    case = 2
      call qprintc(wrow%(LI),wcol%(LI),wcol%(LI)+wcols%(LI)-1,_
                   title$,wattr%(LI))
    rem UpperRight
    case = 3
      call qprint(wrow%(LI),wcol%(LI)+wcols%(LI)-len(title$)-2,_
                  title$,wattr%(LI))
    rem LowerLeft
    case = 4
      call qprint(wrow%(LI)+wrows%(LI)-1,wcol%(LI)+2,title$,wattr%(LI))
    rem LowerCenter
    case = 5
      call qprintc(wrow%(LI)+wrows%(LI)-1,wcol%(LI),wcol%(LI)+wcols%(LI)-1,_
                   title$,wattr%(LI))
    rem LowerRight
    case = 6
      call qprint(wrow%(LI)+wrows%(LI)-1,wcol%(LI)+wcols%(LI)-len(title$)-2,_
                  title$,wattr%(LI))
    case else
  end select
end sub

sub RemoveWindow static
shared Wrow%(),WCol%(),WRows%(),Wcols%(),Wattr%(),WShdw%(),Scrn%(),Wptr(),LI
  if LI = 0 then
    print "NO WINDOW TO REMOVE"
  else
    select case WShdw%(LI)
    case = 1
      call qrest(Wrow%(LI),WCol%(LI)-2,WRows%(LI)+1,WCols%(LI)+2,_
                 Scrn%(Wptr(LI)))
    case = 2
      call qrest(WRow%(LI),WCol%(LI),WRows%(LI)+1,WCols%(LI)+2,Scrn%(Wptr(LI)))
    case else
```

```
      call qrest(WRow%(LI),Wcol%(LI),WRows%(LI),WCols%(LI),Scrn%(Wptr(LI)))
    end select
    LI = LI -1
  end if
end sub

sub Qbox(Row%,Col%,Rows%,Cols%,attr%,BrdrSel%) static
  if rows%>2 and cols%>2 then
    if brdrsel% > 0 and brdrsel% < 6 then
      on brdrsel% gosub single,double,mixed12,mixed21,doubleleftarrow
      call qprint(row%,col%,tl$,attr%)
      call qfill (row%,col%+1,1,cols%-2,asc(th$),attr%)
      call qprint(row%,col%+cols%-1,tr$,attr%)
      call qfill (row%+1,col%,rows%-2,1,asc(lv$),attr%)
      call qfill (row%+1,col%+cols%-1,rows%-2,1,asc(rv$),attr%)
      call qprint(row%+rows%-1,Col%,bl$,attr%)
      call qfill (row%+rows%-1,Col%+1,1,cols%-2,asc(bh$),attr%)
      call qprint(row%+rows%-1,col%+cols%-1,br$,attr%)
      call qfill (row%+1,col%+1,rows%-2 ,cols%-2,asc(" "),attr%)
    else
      call qfill (row%,col%,rows%,cols%,asc(" "),attr%)
    end if
  end if
  exit sub

Single:
  TL$=CHR$(218):TH$=CHR$(196):TR$=CHR$(191)
  LV$=CHR$(179):RV$=CHR$(179)
  BL$=CHR$(192):BH$=CHR$(196):BR$=CHR$(217)
  Return
Double:
  TL$=CHR$(201):TH$=CHR$(205):TR$=CHR$(187)
  LV$=CHR$(186):RV$=CHR$(186)
  BL$=CHR$(200):BH$=CHR$(205):BR$=CHR$(188)
  Return
Mixed12:
  TL$=CHR$(214):TH$=CHR$(196):TR$=CHR$(183)
  LV$=CHR$(186):RV$=CHR$(186)
  BL$=CHR$(211):BH$=CHR$(196):BR$=CHR$(189)
  Return
Mixed21:
  TL$=CHR$(213):TH$=CHR$(205):TR$=CHR$(184)
  LV$=CHR$(179):RV$=CHR$(179)
  BL$=CHR$(212):BH$=CHR$(205):BR$=CHR$(190)
  Return
DoubleLeftArrow:
  TL$=CHR$(17):TH$=CHR$(205):TR$=CHR$(187)
  LV$=CHR$(186):RV$=CHR$(186)
  BL$=CHR$(200):BH$=CHR$(205):BR$=CHR$(188)
  Return
end sub
```

```
sub ClearWindow static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
   call qfill(wrow%(LI)+1,wcol%(LI)+1,wrows%(LI)-2,wcols%(LI)-2,asc(" "),_
              wattr%(LI))
end sub

sub PrtWindow(row%,col%,StrDat$) static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
   call qprint(wrow%(LI)+row%,wcol%(LI)+col%,StrDat$,wattr%(LI))
end sub

sub PrtCWindow(row%,StrDat$) static
shared wrow%(),wcol%(),wrows%(),wcols%(),wattr%(),LI
   call qprintc(wrow%(LI)+row%,wcol%(LI),wcol%(LI)+wcols%(LI),StrDat$,_
                wattr%(LI))
end sub

def fnattr%(fore%,back%)
  local temp%
  temp%=(back%*16)+fore%
  if fore%>15 then temp% = temp% + 112
  fnattr% = temp%
end def

SUB QPRINT INLINE
  $INLINE "QPRINT.BIN"
END SUB
rem CALL QPRINT(ROW%,COL%,STR$,ATTR%)

SUB QPRINTC INLINE
  $INLINE "QPRINTC.BIN"
END SUB
rem CALL QPRINTC(ROW%,COLL%,COLR%,STRDAT$,ATTR%)

SUB QFILL INLINE
  $INLINE "QFILL.BIN"
END SUB
rem CALL QFILL(ROW%,COL%,ROWS%,COLS%,CHAR%,ATTR%)

SUB QATTR INLINE
  $INLINE "QATTR.BIN"
END SUB
rem CALL QATTR(ROW%,COL%,ROWS%,COLS%,ATTR%)

SUB QSAVE INLINE
  $INLINE "QSAVE.BIN"
END SUB
rem CALL QSAVE(ROW%,COL%,ROWS%,COLS%,SCRN%(??))

SUB QREST INLINE
  $INLINE "QREST.BIN"
END SUB
```

```
rem CALL QREST(ROW%,COL%,ROWS%,COLS%,SCRN%(??))
'--------------------------------------------------------------------------------
'end of PBAHELP.BAS
'--------------------------------------------------------------------------------
```

Appendix A.11
Assembler Routines

```
'------------------------------------------------------------------
'These utilities are released into the public domain for anyone
'to use as they see fit; however, anyone who feels inspired and
'would like to contribute to my printer paper and computer
'supplies fund may feel free to do so.
'Rick Fothergill
'141 Oak Hill Road
'Pittsfield, MA 01201
'(413) 442-2456
'GEnie Address: R.FOTHERGILL
'Compuserve: 76210,443
'..................................................................
```

The Q*.ASM routines are used as part of PBAFEDIT.BAS and PBAHELP.BAS programs.

Appendix A.11.1
QATTR.ASM

```
            title         QATTRS
            page          60,132

attr        equ           [bp + 06h]
cols        equ           [bp + 0Ah]
rows        equ           [bp + 0Eh]
col         equ           [bp + 12h]
row         equ           [bp + 16h]

program     segment
            assume        cs:program
            push          bp
            mov           bp,sp
            push          ds
            push          es
            mov           ah,15
            int           10h
            cmp           al,7
            jnz           color
            mov           bx,0B000h
            mov           dx,03BAh
            jmp           short ok
color:      mov           bx,0B800h
            mov           dx,03DAh
ok:         push          bx
            pop           es
            lds           si,rows
            mov           cx,ds:[si]
            push          cx
            push          dx
            xor           bx,bx
            mov           bl,ah
            lds           si,row
            mov           ax,ds:[si]
            dec           ax
            mul           bx
            shl           ax,1
            lds           si,col
            mov           bx,ds:[si]
            dec           bx
            shl           bx,1
            add           bx,ax
            mov           di,bx
            lds           si,attr
            mov           bx,ds:[si]
            cld
            lds           si,cols
            mov           si,ds:[si]
            pop           dx
nxtrow1:    push          di
            cli
            mov           cx,si
```

```
doagain1:       inc             di
waitlo1:        in              al,dx
                test            al,01
                jnz             waitlo1
waithi1:        in              al,dx
                test            al,01
                jz              waithi1
                mov             al,bl
                stosb
                loop            doagain1
                sti
                pop             di
                pop             cx
                dec             cx
                jcxz            endit
                push            cx
                add             di,160
                jmp             short nxtrow1
endit:          pop             es
                pop             ds
                pop             bp
program         ends
                end
```

Appendix A.11.2
QFILL.ASM

```
                title           QFILLS
                page            60,132

attr            equ             [bp + 06h]
char            equ             [bp + 0Ah]
cols            equ             [bp + 0Eh]
rows            equ             [bp + 12h]
col             equ             [bp + 16h]
row             equ             [bp + 1Ah]

program         segment
                assume          cs:program
                push            bp
                mov             bp,sp
                push            ds
                push            es
                mov             ah,15
                int             10h
                cmp             al,7
                jnz             color
                mov             bx,0B000h
                mov             dx,03BAh
                jmp             short ok
color:          mov             bx,0B800h
                mov             dx,03DAh
ok:             push            bx
                pop             es
                lds             si,rows
                mov             cx,ds:[si]
                push            cx
                push            dx
                xor             bx,bx
                mov             bl,ah
                lds             si,row
                mov             ax,ds:[si]
                dec             ax
                mul             bx
                shl             ax,1
                lds             si,col
                mov             bx,ds:[si]
                dec             bx
                shl             bx,1
                add             bx,ax
                mov             di,bx
                lds             si,char
                mov             ax,ds:[si]
                mov             bl,al
                lds             si,attr
                mov             ax,ds:[si]
                mov             ah,al
                cld
                lds             si,cols
```

```
        mov     si,ds:[si]
        pop     dx
nxtrow1: push   di
        mov     cx,si
        cli
waitlo1: in     al,dx
        test    al,01
        jnz     waitlo1
waithi1: in     al,dx
        test    al,01
        jz      waithi1
        mov     al,bl
        stosw
        loop    waitlo1
        sti
        pop     di
        pop     cx
        dec     cx
        jcxz    endit
        push    cx
        add     di,160
        jmp     short nxtrow1
endit:  pop     es
        pop     ds
        pop     bp
program ends
        end
```

Appendix A.11.3
QPRINT.ASM

```
              title     QPRINTS
              page      60,132

Attr          equ       [bp + 06h]        ; ATTR%
StrDat        equ       [bp + 0Ah]        ;
Col           equ       [bp + 0Eh]        ;
Row           equ       [bp + 12h]        ;

program       segment
              assume         cs:program
              push           bp
              mov            bp,sp
              push           ds
              push           es
              les            di,StrDat
              mov            cx,es:[di]
              and            cx,7FFFh
              jcxz           endit
              push           cx
              mov            dx,ds:[0]
              push           dx
              mov            ah,15
              int            10h
              cmp            al,7
              jnz            color
              mov            bx,0B000h
              mov            dx,03BAh
              jmp            short ok
color:        mov            bx,0B800h
              mov            dx,03DAh
ok:           push           bx
              pop            es
              push           dx
              xor            bx,bx
              mov            bl,ah
              lds            si,row
              mov            ax,ds:[si]
              dec            ax
              mul            bx
              shl            ax,1
              lds            si,col
              mov            bx,ds:[si]
              dec            bx
              shl            bx,1
              add            bx,ax
              mov            di,bx
              lds            si,attr
              mov            bx,ds:[si]
              lds            si,strdat
              mov            si,ds:[si + 2]
              pop            dx
              pop            ds
```

```
                pop             cx
                cld
                cli
waitlo1:        in              al,dx
                test            al,01
                jnz             waitlo1
waithi1:        in              al,dx
                test            al,01
                jz              waithi1
                movsb
                mov             es:[di],bl
                inc             di
                loop            waitlo1
                sti
endit:          pop             es
                pop             ds
                pop             bp
program         ends
                end
```

Appendix A.11.4
QPRINTC.ASM

```
                title   QPRINTCS
                page    60,132

Attr            equ     [bp + 06h]      ; ATTR%
StrDat          equ     [bp + 0Ah]      ;
ColR            equ     [bp + 0Eh]      ;
ColL            equ     [bp + 12h]      ;
Row             equ     [bp + 16h]      ;

program         segment
                assume          cs:program
                push            bp
                mov             bp,sp
                push            ds
                push            es
                les             di,StrDat
                mov             cx,es:[di]
                and             cx,7FFFh
                jcxz            endit
                push            cx
                mov             dx,ds:[0]
                push            dx
                mov             ah,15
                int             10h
                cmp             al,7
                jnz             color
                mov             bx,0B000h
                mov             dx,03BAh
                jmp             short ok
color:          mov             bx,0B800h
                mov             dx,03DAh
ok:             push            bx
                pop             es
                push            dx
                xor             bx,bx
                mov             bl,ah
                lds             si,row
                mov             ax,ds:[si]
                dec             ax
                mul             bx
                shl             ax,1
                lds             si,colL
                mov             bl,ds:[si]
                lds             si,colR
                mov             bh,ds:[si]
                add             bl,bh
                xor             bh,bh
                shr             bx,1
                shr             cx,1
                sub             bx,cx
                dec             bx
                shl             bx,1
```

```
                add             bx,ax
                mov             di,bx
                lds             si,attr
                mov             bx,ds:[si]
                lds             si,StrDat
                mov             si,ds:[si+2]
                pop             dx
                pop             ds
                pop             cx
                cld
                cli
waitlo1:        in              al,dx
                test            al,01
                jnz             waitlo1
waithi1:        in              al,dx
                test            al,01
                jz              waithi1
                movsb
                mov             es:[di],bl
                inc             di
                loop            waitlo1
                sti
endit:          pop             es
                pop             ds
                pop             bp
program         ends
                end
```

Appendix A.11.5
QREST.ASM

```
                title       QREST
                page        60,132

source          equ         [bp + 06h]
cols            equ         [bp + 0Ah]
rows            equ         [bp + 0Eh]
col             equ         [bp + 12h]
row             equ         [bp + 16h]

program         segment
                assume      cs:program
                push        bp
                mov         bp,sp
                push        ds
                push        es
                mov         ah,15
                int         10h
                cmp         al,7
                jnz         color
                mov         bx,0B000h
                mov         dx,03BAh
                jmp         short ok
color:          mov         bx,0B800h
                mov         dx,03DAh
ok:             push        bx
                pop         es
                lds         si,rows
                mov         cx,ds:[si]
                push        cx
                push        dx
                xor         bx,bx
                mov         bl,ah
                lds         si,row
                mov         ax,ds:[si]
                dec         ax
                mul         bx
                shl         ax,1
                lds         si,col
                mov         bx,ds:[si]
                dec         bx
                shl         bx,1
                add         bx,ax
                mov         di,bx
                lds         si,cols
                mov         bx,ds:[si]
                lds         si,source
                cld
                pop         dx
nxtrow1:        push        di
                cli
                mov         cx,bx
waitlo1:        in          al,dx
```

```
            test        al,01
            jnz         waitlo1
waithi1:    in          al,dx
            test        al,01
            jz          waithi1
            movsw
            loop        waitlo1
            sti
            pop         di
            pop         cx
            dec         cx
            jcxz        endit
            push        cx
            add         di,160
            jmp         short nxtrow1
endit:      pop         es
            pop         ds
            pop         bp
program     ends
            end
```

Appendix A.11.6
QSAVE.ASM

```
                title           QSAVE
                page            60,132

dest            equ             [bp + 06h]
cols            equ             [bp + 0Ah]
rows            equ             [bp + 0Eh]
col             equ             [bp + 12h]
row             equ             [bp + 16h]

program         segment
                assume          cs:program
                push            bp
                mov             bp,sp
                push            ds
                push            es
                mov             ah,15
                int             10h
                cmp             al,7
                jnz             color
                mov             bx,0B000h
                mov             dx,03BAh
                jmp             short ok
color:          mov             bx,0B800h
                mov             dx,03DAh
ok:             push            bx
                pop             ds
                les             di,rows
                mov             cx,es:[di]
                push            cx
                push            dx
                xor             bx,bx
                mov             bl,ah
                les             di,row
                mov             ax,es:[di]
                dec             ax
                mul             bx
                shl             ax,1
                les             di,col
                mov             bx,es:[di]
                dec             bx
                shl             bx,1
                add             bx,ax
                mov             si,bx
                les             di,cols
                mov             bx,es:[di]
                les             di,dest
                cld
                pop             dx
nxtrow1:        push            si
                cli
                mov             cx,bx
waitlo1:        in              al,dx
```

```
                test        al,01
                jnz         waitlo1
waithi1:        in          al,dx
                test        al,01
                jz          waithi1
                movsw
                loop        waitlo1
                sti
                pop         si
                pop         cx
                dec         cx
                jcxz        endit
                push        cx
                add         si,160
                jmp         short nxtrow1
endit:          pop         es
                pop         ds
                pop         bp
program         ends
                end
```

Bibliography

1. AASHTO; "AASHTO Guide for the Design of Pavement Structures, 1986". American Association of State Highway and Transportation Officials, Washington D.C., 1986.

2. Ordonez, G. and Vinson, T.; "Effect of Environmental Factors on Pavement Deterioration, Volume I". Transportation Research Institute, Department of Civil Engineering, Oregon State University, Corvallis, Oregon, 1988.