18

GOVERNMENT OF BANGLADESH
MINISTRY OF ENVIRONMENT AND FORESTS

SYSTEMS ANALYSIS

FORESTRY MASTER PLAN

ASIAN DEVELOPMENT BANK (TA NO. 1355-BAN)

UNDP/FAO BGD 88/025

1992

GOVERNMENT OF BANGLADESH
MINISTRY OF ENVIRONMENT AND FORESTS

SYSTEMS ANALYSIS

FORESTRY MASTER PLAN

ASIAN DEVELOPMENT BANK (TA NO. 1355-BAN)

UNDP/FAO BGD 88/025

1992

PROJECT 372001/25
FORESTRY MASTER PLAN,
BANGLADESH (TA NO.1355-BAN)

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 05 NOVEMBER 1992

SYSTEMS ANALYSIS

RIMS

Introduction

The Resource Management System(RMS) had been designed and developed as a comprehensive forest management planning system under the FAO/UNDP Project BGD/79/017, Assistance to the Forestry Sector, Bangladesh. Establishment of the computerized RMS was undertaken in mid eighties and became fully operational in 1988. Computer operations of this system consisted of data management, processing and reporting to accumulate, store and analyze data from forest inventories, silvicultural prescriptions, growth and yield information, socio-economic factors and economic information at various levels of aggregation.

RMS computer operation was originally set on two levels covered by macro and micro model. The micro model was designed for information processing at a mappable basic management information unit level termed as Discrete Landuse Unit(DLU) and the macro model for providing information at a higher level, usually the Forest Division.

Macro Model RMS - The macro model consisted of mainly spreadsheets developed with "AppleWorks" software package on the Apple IIe computers and termed as RMS Management Model Spreadsheets and was designed for general wood resource management planning, country wide wood supply projections and strategic planning. These spreadsheets provided in a table format an accounting of the current and future forest landbase established on the summarized inventory data and management programme. Two types of spreadsheets made up the Management Model RMS:

a. Short Rotation Plantation - accounted for landbase devoted to the following categories of plantations:

Domestic Wood - Plantations with an 18 year cutting cycle with coppice cuts at 6 and 12 years.

Peeler/Fuelwood - Plantations of Albizia moluccana with a 12 year cutting cycle and a thinnings at 5 and 8 years of age.

Short Rotation - Plantations with an 18 year cutting cycle with thinnings at 8 and 12 years of age.

b. Long Rotation Plantation - accounted for the Natural Forest and the Long Rotation Industrial Plantations.

It was recommended that the spreadsheets should be updated and new ones developed for other areas as more information is made available and different management programmes are to be evaluated. But with the obsoletion of Apple IIe and it's AppleWork and since no further development and use was made for this Management Model Spreadsheets, they are no longer in use or available in the computers of RMS.

Micro Model RMS - This level of computer operation stores, updates, processes, and retrieves information primarily on individual units of forest land called Sub-block or DLU. Sub-blocks are pre-identified and mappable blocks of more or less homogenous tree crop which can be subjected to determined management practices and provide the basis for operational, area related forest management planning and control. This is also an integration of two separate computer programmes:

PROJECT 372001/25
FORESTRY MASTER PLAN,
BANGLADESH (TA NO.1355-BAN)

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 5 NOVEMBER 1992

SYSTEMS ANALYSIS

TABLE OF CONTENTS

a. Sub-block inventory Programmes - developed in Dbase III plus programming language under FAO/UNDP Project BGD/79/017. This programme is made up of programmes and datafiles to enter and compile inventory and DLU data. This is essential for creating input data files for modelling programme.

b. RMS Modelling Programmes - is a computer package in BASIC programming language developed originally in the period 1984-1986 for standwise forest management planning and implemented in Sri Lanka under FRDP, carried out by GOSL/WB/FINNIDA. This package was adapted and modified to meet the objectives of the Bangladesh RMS. This programme takes the output of the inventory programme as input and produces the silvicultural prescriptions and cutting programmes as output, therefore the operation of this programme can be described as:

   - DLU specific information (area, species, DBH, height etc) processing.
   - Calculation of growth and volume functions for each species group.
   - Determining standard silvicultural prescriptions and cutting programmes for each species group and management objectives.

This micro model RMS has been used for the preparation of long and short term management plans for Cox's Bazar, Chittagong, Sylhet and four Coastal Afforestation areas. Currently this micro model portion is under operation and now termed as Resource Information Management System(RIMS) and henceforth any mention of RIMS in this report will refer to this Micro Model RMS.

General Description

To optimize wood production in forest plantations this modelling programme was designed to be a "Standwise Management System". Thus, the smallest planning unit (for which individual treatment would have to be applied) is a more or less homogeneous stand that is characterized mainly by it's species, age site class (height above age) and stocking degree and would thus require a different management regimen as it's neighbouring stands in order to realize it's full wood volume and timber quality production potential.

RIMS is a combination of Dbase and BASIC programme and data files. It's inventory portion is in Dbase III plus and modelling portion is in a combination of Dbase and BASIC programmes.

1. Inventory Programme

The inventory management programme of the RIMS, termed as "Dbase RMS", was developed in Dbase programming language to enter and compile the inventory and DLU data so that this data could be transferred to the RMS Modelling Programme as input. There are two general types of RMS area data and two corresponding datafile types:

a. Inventory datafiles - contains the forest inventory stratum-wise data.
b. -DLU datafiles - contains information of the mappable forest management units.

The DLU data file is furnished with the following sub-block information:

- Identifying codes of forest Division, District, Range, Beat, Block and Sub-block.
- Land capability, Land use, Area, Altitude, Soil type, slope, Inventory year.
- Species code and their percentage in order of dominance, Plantation year.
- Planting method, Out crop, Purpose of planting, Controlling authority.
- Previous treatment period, Treatment type, Condition, Failure reason, Natural regeneration, Coppicing, Regenerated/coppiced main species.

- Damage of plantation, Type of damage, Damage degree, Damage stem height.
- Distribution of remaining trees.
- Stem density, Stem height, Stand height, Basal area, DBH.
- Map index.

These information is recorded in database files of respective operational division according to the information source classification in three main types of data file group:

- Inventory data
- Extrapolated data
- Book statistic data

Output from the inventory programme are as follows:

- Generation of input data file in a format conforming to the modelling programme requirements.
- A listing of the raw inventory or DLU data. Area total are given by Forest Range and Division.
- Report of calculated total area by Block, Range and Division of the four landuse categories: Natural Forests, Plantations, Denuded Areas and Agriculture
- Report with calculation of total area in each plantation inventory age group by Forest Block, Range and Division.
- Report with calculation of total area in each plantation unit age group by forest block, Range and Division.
- Check report of the inventory data for coding errors or missing data and lists the incorrect DLUs.
- Check report of the DLU data for any coding errors or missing data.

2. RMS Modelling Programme

Basing upon the basic information provided by the input inventory data file regarding present situation and calculating growth and volume function for each species group, the management plans are compiled. The modelling programme determines the management plan for individual DLU for the total rotation period by generating silvicultural and cutting prescriptions.

Standard Silvicultural Treatment Programme - For each main species (or species group) a computerized treatment programme is adapted and standardised to allow fast and unbiased treatment prescription for each stand (or species, stocking and age class). The proposed treatments depend on various stand parameters, such as:

- tree species
- stocking/ha
- stand conditions (number of damaged trees/ha)
- site class

The most important operations prescribed for plantations are:

- clearings of understocked young stand and re-planting.
- vacancy planting.
- weeding.
- cleaning and creeper cutting.
- spacing.
- singling (Teak).
- early thinnings to stimulate growth.
- remove undesired invading species and weed and to favour the elite trees at an early stage.

Cutting Prescriptions - This model prescribes future cutting operations for each individual stand (sub-block) for the whole rotation period. The model predicts the timing of thinnings and regeneration cuttings, the number of stems and basal area, to be removed and maintained, the average DBH and the average height of removals and remaining crop and most of all, the volumes (total and log volume) of the removals.

Before any projections until the rotation is end are activated the model checks if the stand is reasonably stocked (minimum stocking above age) to be maintained or if it should be cleared and replanted (heavily understocked areas).

If a stand is qualified to be maintained the model then compares the present stocking with desired stocking (given by yield tables as well as thinning intervals) by using number of trees/ha and basal area/ha as parameters. Cutting is prescribed when the stocking exceeds a desired stocking by a certain percentage.

The model projects the DBH and height for future thinning and the main crop by using individual growth functions. If the present DBH or height is above or below the yield table values, this is expected to prevail in the future too and the projected yields will be consequently above or below the normal yields (yield table). However, since the stands will be re-inventoried after each operation the actual state of the stand will be known and thus the projection will become more accurate. The model, like any cutting prediction, is sensitive to age, mean height, stocking/ha, and thus only as good as the quality of the field data will be.

Operational Records - The operations carried out in a stand (sub-block) is reported on a special field form to the Data Processing Unit where they are registered. For each sub-block the silvicultural and cutting operation areas, the removed volumes and the labour requirement is calculated.

Information Compiled and outputs - The output of the modelling program can be classified into following three major categories:

- Silvicultural prescriptions
- Prediction of growth and yield
- Choice of management options

Major listings from the programme are:

- Sub-block standwise description and silvicultural prescriptions.
- Beatwise 5-year operation planning on each sub-block with a Beat and Range wise summary.
- Annual silvicultural treatment; Area summary and the cutting potential summary.
- Average annual cutting-area and wood volume by species cutting types, utilization group (whole rotation).
- Average annual silvicultural treatment-areas by type of treatment (for the whole rotation).
- Stand development class distribution (established, non-sampling stage, thinning stage & mature stage).
- Annual resource requirement for silvicultural operations i.e. weeding, cleaning etc.
- Annual resource requirement for cutting operation i.e. thinning and final felling.
- Volume, area and resource summary listing on performed operation.
- Area, growing stock, age class distribution, separately for short and long rotation species.

Computer Programmes

1. Inventory Programme

Computer operation of the RMS inventory programme involves the following Dbase programme files:

File Name	Operation
CAFF.PRG	- Coastal Afforestation data
CAFREP.PRG	- Report, Coastal Afforestation data
CHKCAF.PRG	- Checking, Coastal Afforestation data
CHKDLU.PRG	- Checking, DLU data
CHKINV.PRG	- Checking, Inventory data
MACREP.PRG	- Report
MICREP.PRG	- Report
MICREP1.PRG	- Report, Inventory/ Dlu Data
MICREP2.PRG	- Report, Forest landuse Summary
MICREP3.PRG	- Report, Inventory Age Profile
MICREP4.PRG	- Report, DLU Age Profile
MICREP5.PRG	- Report, Check Inventory Data
MICREP6.PRG	- Report, Check DLU Data
RMS.PRG	- Main Programme
VOLCALC.PRG	- Volume calculation

Other programme modules in use are:

RMSLIB.PRG, SPLIST.PRG, SPLIST2.PRG, TEMPDLU.PRG, APPDLU.PRG, MACRO.PRG, MICAPP.PRG, MICAPP2.PRG, MICRO.PRG, NFTYPE.PRG, PRINTER.PRG, REP1.PRG, REP1B.PRG, MICREP2*.PRG, MICREP4*.PRG, MACREP3.PRG, CAFREP2.PRG

2. Modelling Programme

Computer operation of the RMS modelling program involves the following Dbase and BASIC programme files:

File Name	Operation
Dbase programme files	
BATCHII.PRG	- Selects "Batch", Subroutine to BATCHSEL.PRG
BATCHIN.PRG	- Calls back deleted data
BATCHOUT.PRG	- calls out data for volume, silvicultural, cutting projections
BATCHSEL.PRG	- Creates sequential datafile for BASIC programme input
BATCHTRA.PRG	- Transfers old data records to backup data base file
COMPMENU.PRG	- Main menu for Dbase programme
DIVISEL.PRG	- Create division wise external data file for BASIC programme
INPOP.PRG	- Programme subroutine
INPUT.PRG	- Programme subroutine
PURINDEX.PRG	- Creates index files for a given index key
PURUPDAT.PRG	- Updating and entering new data
TRANMANI.PRG	- Transfer information between data files

Other programme modules in use are:

INVINDEX.PRG, PRINTER.PRG

BASIC Programme files

MAINMENU.BAS	-	Main menu programme
MULTMENU.BAS	-	Multi division selection menu programme
DEVCLDRA.BAS	-	Range wise listing of stand development classes
RESOANA.BAS	-	Annual resources for silvicultural and cutting operations
PUROPE.BAS	-	Location, area and volume for a given period and operation
PURPOPE.BAS	-	Summary of DLU and Beat operations, labour requirements
PRINTER.BAS	-	Initializes printer
PTEXTMNU.BAS	-	Text data creating and listing menu programme
TEXTENT.BAS	-	Provide facility to enter, edit and update data to file
TEXTLIST.BAS	-	Lists the text data
PURMENU.BAS	-	Sequential data file indexing menu programme
VOLCALC.BAS	-	Calculates volume per hectare for each DLU
PROADD.BAS	-	Forecast future silvicultural treatments, cutting operations
PROGLIS1.BAS	-	Five year silvicultural and cutting operation listing
TREATSUM.BAS	-	Listing of silvicultural treatment areas and cutting potentials
ARGRSTRA.BAS	-	Provides range wise listing of each age group of each species
PURRASUM.BAS	-	List the range wise treatment operation areas
COUNRBBS.BAS	-	Lists no. of Ranges, Beats, Blocks, DLUs in a Division

Other programme modules in use are:

ARGRMENU.BAS, LOSPCODE.BAS, TM.BAS, TRETMENU.BAS, PROMENU.BAS

System Evaluation

RIMS is a vital part of the Resource Management System in the context of resource information storage and manipulation, planning, decision making and determination of suitable management practice standards. Figure 1 illustrates the position of RIMS in the Resource Management System. Realizing it's utility and future role in the RMS, Dr. D.R.Pelz and Mr.M.Pushparajah, FAO Consultant, Project BGD/85/085, commented in the Mission Report titled "Review of the Forest Resources Management System" (July 05 - July 23, 1987):

> "At present the computer programmes used are applied mainly for plantation management. It is designed as flexible and dynamic system that can be easily updated and improved as information becomes available and the data base is increased. The use of the system can be expanded to all areas for plantation management, provided the input data are available.
>
> In addition to increasing the territorial coverage of the system it is strongly recommended to develop the system further to a comprehensive forest data base system that can support both management and policy decisions at various levels.
>
> Information on socio-economic data, financial analysis data, and other data referable to the geographic units should be included."

Conceding the expectations of the FAO Consultants, the system is evaluated focusing on the optimization of the present facilities in the context of utilization, operation and development.

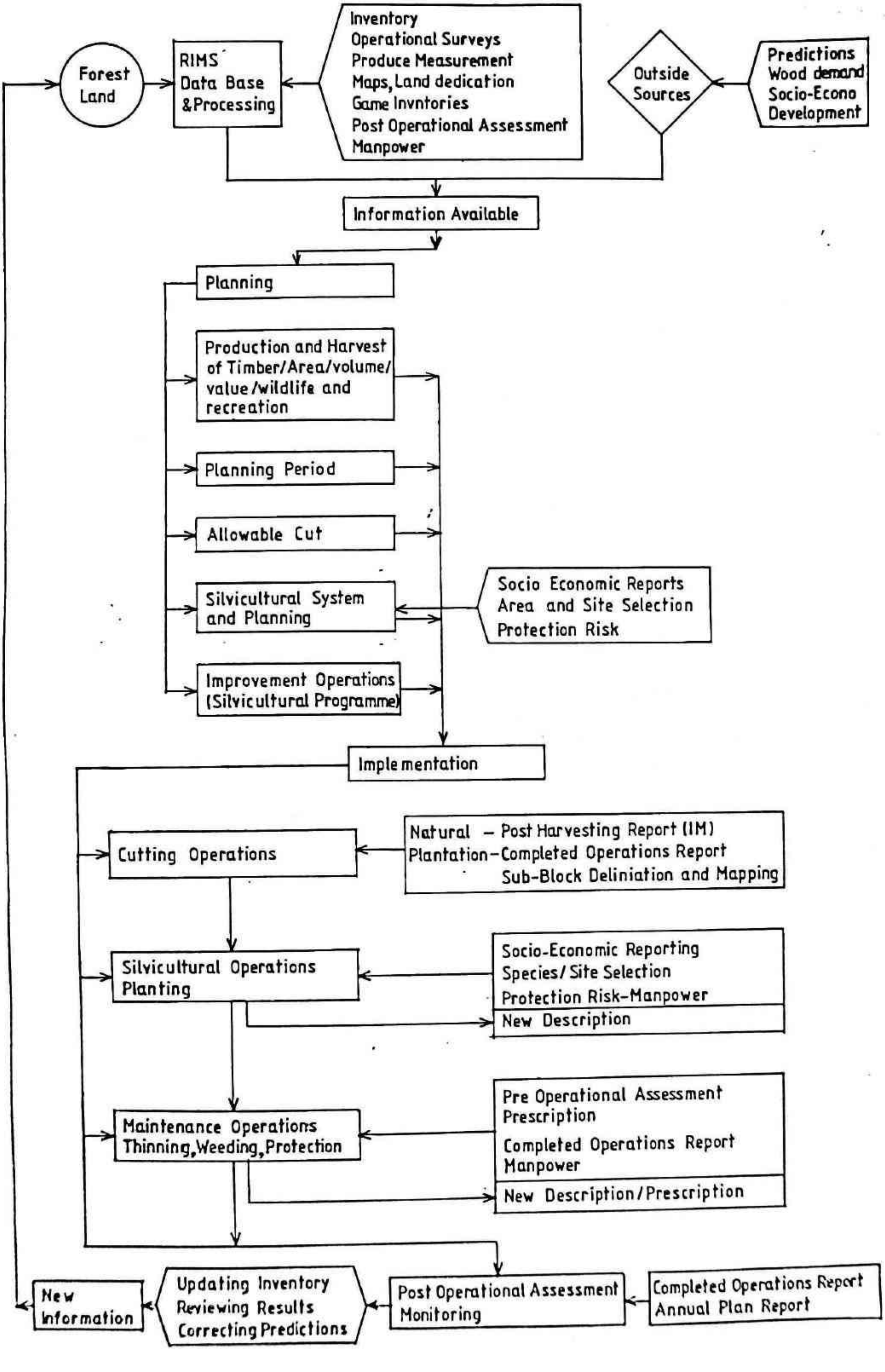
Forest Land
RIMS Data Base &Processing
Inventory
Operational Surveys
Produce Measurement
Maps, Land dedication
Game Invntories
Post Operational Assessment
Manpower
Outside Sources
Predictions
Wood demand
Socio-Econo
Development
Information Available
Planning
Production and Harvest of Timber/Area/volume/value/wildlife and recreation
Planning Period
Allowable Cut
Silvicultural System and Planning
Socio Economic Reports
Area and Site Selection
Protection Risk
Improvement Operations (Silvicultural Programme)
Implementation
Cutting Operations
Natural – Post Harvesting Report (IM)
Plantation – Completed Operations Report
Sub-Block Deliniation and Mapping
Silvicultural Operations Planting
Socio-Economic Reporting
Species/Site Selection
Protection Risk-Manpower
New Description
Maintenance Operations Thinning, Weeding, Protection
Pre Operational Assessment
Prescription
Completed Operations Report
Manpower
New Description/Prescription
New Information
Updating Inventory
Reviewing Results
Correcting Predictions
Post Operational Assessment
Monitoring
Completed Operations Report
Annual Plan Report

Figure 1 - RIMS in Forest Management System

1. Utilization

RIMS is installed at the Forest Department Head Quarters, Dhaka and presently used for preparation of long and short term management plans for Cox's Bazar, Chittagong, Sylhet and four Coastal Afforestation areas. Although this system is working to serve present purpose with difficulty the constraints to be overcome to further it's utilization are as follows:

- RIMS is a "standwise" system, that means it is designed mainly for plantations of more or less homogeneous species. Therefore it is difficult to implement for natural forests or any other kind of plantations.

- It requires initial field inventory data at the level of discrete working units but this data is not always available and accuracy of this data affects the programme output, physical reference etc.

- The programme presently works with only 14 species group and inclusion of any new species group is difficult.

- The programme is embedded with parameters for which no user entry or editing facility is provided but need frequent modification. These parameters are under the following options/ operations:

  - Volume functions.
  - Species group selection/ introduction.
  - No. of stems/ ha to determine understocking.
  - Remaining stems/ ha for each species group and cutting operation.
  - Maximum no. of stems/ ha for each species group and cutting operation.
  - Standard cutting ages for each species group and cutting operation.
  - Minimum and maximum ages for each treatment and species group.
  - Minimum and maximum n/ ha for each treatment and species group.
  - Minimum and maximum DBH for each treatment and species group.
  - Projection year for each treatment and species group.
  - Frequency of area for each treatment and species group.
  - Minimum required basal area index to qualify for 2nd and 3rd thinning.
  - DBH and Height projection parameters.
  - Time period (starting and ending).
  - Other parameters used as constant in the programme but need modification.

- Some modifications required with the change of management practices cannot be incorporated in RIMS. For example: old concept of growth model, thinning regimes, rotation fixation etc.

- Introduction of a new forest division under RIMS operation is not easily attainable and needs much of data and parameter modification, calibration, field test confirmation etc.

- Mapping of the units is done manually. There was recommendation for procurement of GIS software named "Comprehensive Resource Inventory and Evaluation" (CRIES) but yet to be obtained.

- RMS deals with different level of information processing but RIMS is still engaged with it's original stage of plantation management and the macro level spreadsheets are no longer in use. Therefore, present RIMS is only a part of the information system and there are more to be developed. Figure 2 shows the information flow.

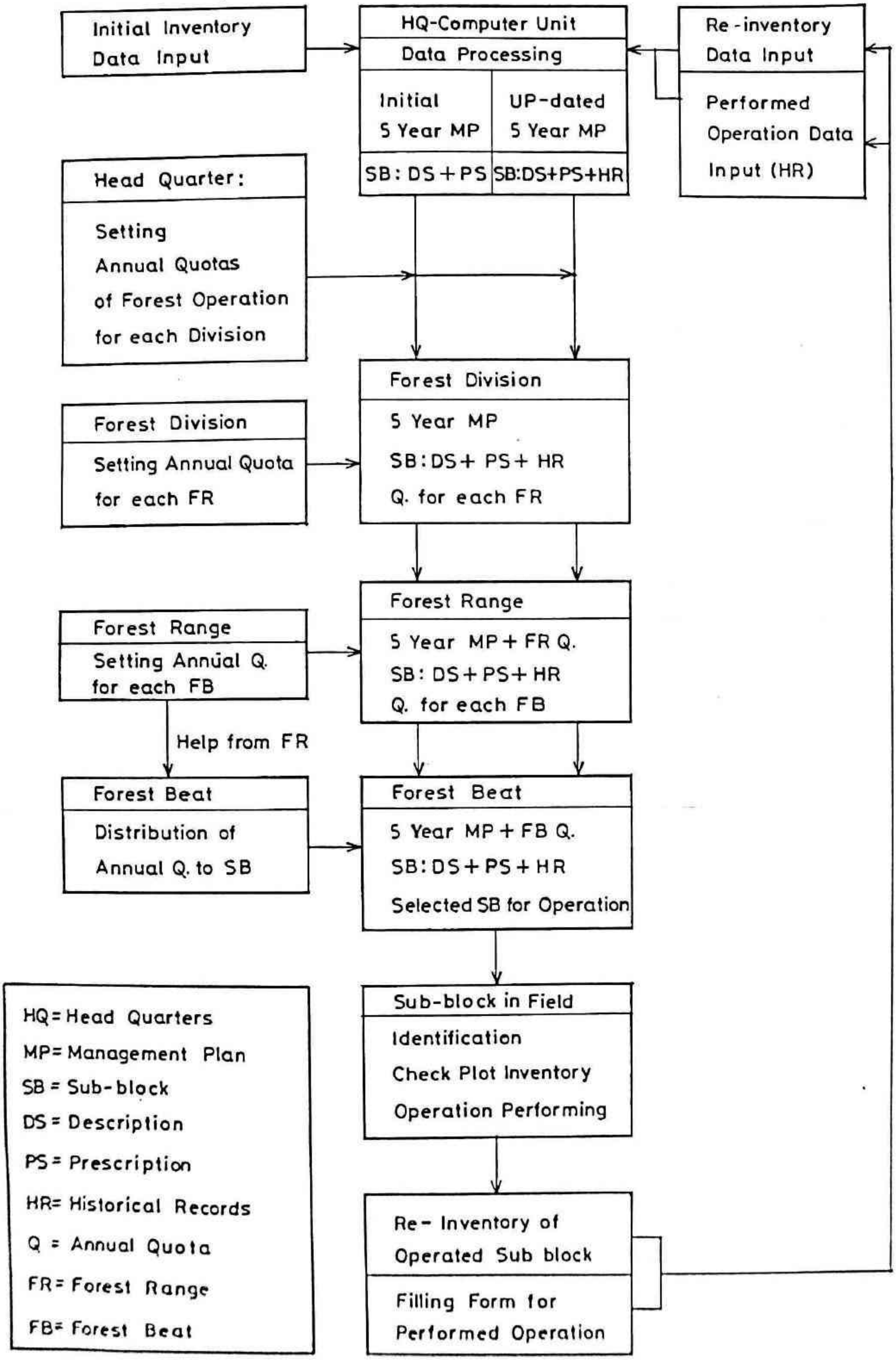

Figure 2 - Flow of Information for RMS Implementation (5 Year Management Cycle)

2. Operation

The computer operation of RIMS is analyzed in the context of user facilities, efficiency, software maintenance and hardware.

User Facilities - User friendliness of a computer system is the measure of it's acceptability and user facilities provided in a system determines the degree of friendliness. RIMS is a complete failure in this regard. Following are some of the hindering features of RIMS:

- User has to load and run multiple sessions of Dbase and BASIC software.
- Data file from one part of programme is to be transferred manually to other parts.
- There is no easy menu navigation system or on line help facility.
- There are descriptive documents but no operation guide for the learner/user.
- No error checking facility at entry level.
- Editing and on screen display facility is very poor.
- Programme crashing and data loss occur without any warning.
- User need to be a Dbase and BASIC programmer with forestry background to get most out of the system.
- Programme activity or data is not transparent enough.

Efficiency - The efficiency of the programme is considered from the point of system optimization possibilities. Concept and design of this programme dates back to 1984 and there are much to consider with respect to the present development of computer systems. Nowadays computer systems run faster and flawlessly. Following are some of the features of RIMS showing inefficiency or need of improvement:

- RIMS programmes and menus are not integrated under one system.
- There are some manual manipulation, which could be done automatically.
- Frequent data file loading, saving and transformation.
- Absence of entry level error checking.
- Difficulty in editing, screen viewing.
- No easy cancel or suspension of individual operations.
- System crashing and data losing.
- Use of delete and recall for record selection.
- Use of filter condition rather than using index files.
- Embedding data in the program rather than in data files.
- Variable parameters put as constants without modification facility.

Software Maintenance - Though there were recommendation for one computer programmer, presently this software is maintained by a person of forestry background with some training on RIMS. With long time association and personal interest he is maintaining the system well but he is not capable for system development and enhancement. Moreover, since he is not recognized as a computer programmer, he might be transferred to other sections at any time as per Govt. rules.

This software needs frequent modifications, but the documentation is not well oriented to enhance easy programme modifications. If a new person comes to operate the system, he will need to go through a long and tedious learning processes.

Hardware - RIMS is installed in one of IBM compatible computer with 80286 processor. Although for present need it's speed is adequate, operations will be much faster with 80386 or 80486 machines. If GIS software is installed it will need mouse, digitizer and plotter as well as faster machine with enough memory to operate.

3. Development

It is already mentioned that present RIMS is only a part of the resource management system and which need much of improvement. As scope of development, need and possibility of improvement and expansion of present system is considered in the context of utilization and software system.

Utilization - Present system is designed only for specific sub-block based plantation management but there are more aspects of the resource management that need computerization. The inventory database organization in RIMS is aimed at the requirement of the silvicultural operations and management. With the help of GIS software this inventory could be developed into an integrated land base information system of forests and forest resources of Bangladesh.

In the context of resource management system the scope of development of RIMS can be described as follows:

- Information maintenance and processing at other levels of resource management.
- Increase of territorial coverage with diversified plantation management facility.
- Development of the system as an analytical tool to support management and policy decisions at various levels. For example, this system could be developed to suggest suitable species group for plantation.
- Inclusion of socio-economic and other data referable to the geographic units.
- Incorporation of financial analysis and data maintenance of plantation.

Software System - In the context of 1984-86 software development standards it was described as a structured, flexible and dynamic system that can be easily updated and improved. But this claim is denied where for a small change user need to modify the programme code and the significant portion of the system is written in BASIC, presently considered to be the worst programming standard. Moreover the technical aspects of the programme is not well documented and for that reason modifications became impossible although the programme code and subroutines are marked and modifications suggested. Therefore, it is implied that redesign of the programme is essential to eliminate the limitations discussed and to make it well suited for present and future requirements.

Present programme code volume is around 10,000 lines, but with structured modular design this volume will reduce to two third of this size. Although the programme is in BASIC, there were attempts to make it structured and subroutines were marked and comments added to code for clarity. Therefore, most of the subroutines can be extracted for modification and converted into other languages. Modification and integration of Dbase portion will need least effort. If GIS software is incorporated, provision for information interchange facility to be provided for geo-referencing of locations and attributes.

Recommendations

RIMS is found to be a software system essential for forest resource management but weak in construction, operation and versatility. Although it seems technically sound for present limited use, rigidity of the system as well as modification and expansion complications impede it's development as a complete information system of resource management. Therefore, considering present status and difficulties discussed, following recommendations are made with the view to future development of the system as a complete and versatile forest information system:

1. **System Design**

   a. Technical aspects to be reviewed and modified to fit present and future resource management strategies. Computer activities to be incorporated with the system in future are to be identified and development provision to be provided accordingly.

   b. Emphasis should be given to possible extraction and conversion of present subroutines and technical details for easier upgrading. Provision for networking facility should be provided with the system to avail the facilities of future communication development and decentralization of RIMS operations.

2. **Programming**

   a. Involvement of the original programmer for short period is desirable for providing programme conversion guidelines and explanation of the technical aspects. Estimated period is about one to two man months.

   b. A local consultant programmer should be engaged for conversion and programming. Provision to be kept so that he can be consulted for future development, time to time modifications, feedback analysis and adjustments. A programmer with forestry background is preferable. Estimated period is about six to eight man months.

   c. The programming team must involve one programmer from Forest Department with forestry background, who will maintain the system after completion of programme.

   d. If none of the programmers is with forestry background, one person having adequate database and computer knowledge with forestry background must be provided from Forest Department for assisting in technical aspects, preferably from ACF level or presently involved with RIMS operation.

3. **Software**

   a. Programme structure and construction must be re-designed with modular design standards using faster and wider database environment and stronger programming language, preferably FoxPro or Clipper.

   b. There should be efforts to eliminate the problems and limitations discussed. Programme and code documentation with explanation of technical aspects, operation guide to be provided.

   c. A GIS software must be procured and incorporated with the system for geo-referencing of management locations and attributes. This should also be used as a geo-referencing system for total forests of Bangladesh.

4. **Hardware**

   a. Incorporation of more areas under RIMS activities and introduction of GIS software will demand for faster machine, more storage and memory capabilities. A 33 MHz 80386 IBM compatible computer with 100 Mbyte Hard disk and 4 Mbyte RAM with Super VGA monitor is the minimum configuration recommended.

   b. Mouse, Digitizer and Plotter should be provided for full utilization of GIS potentials.

SIMULATION PROGRAMME

Introduction

There was a supply and demand simulation programme developed by D.J.Edelman, R.N.Byron and D.M.Mansion under project BGD/78/010 written in Apple BASIC. Initially it was assumed that the simulation model programme could be developed basing upon that. But for poor documentation and unavailability of complete programme codes that idea is discarded and a new programme is developed in Clipper database language. Computational principles and methodologies are derived from Statistician and Forest Management Specialist of the Master Plan team for demand and supply simulations respectively and they are described in their reports.

Programme tree structure, programme code, data file structure and sample output formats are included in Appendix 3.

Layout and Operation

The simulation programme is designed to simulate basing upon some initial base year information and subsequent growth calculation parameters. Most of the parameters are kept as variables with user modification facility to provide maximum flexibility. All the entry, and output calculation are done for seven regions which were derived by dividing the territorial area of Bangladesh into seven suitable zones. Following are the area distribution of the regions:

Region	Districts
NORTH WEST/N.WEST	- Dinajpur, Rangpur, Bogra, Rajshahi and Pabna.
NORTH CENTRAL/N.CENT.	- Dhaka, Tangail, Jamalpur and Mymensingh(part).
WEST	- Kushtia, Jessore, Barisal, Faridpur and Khulna(part).
SOUTH	- Khulna(part) and Patuakhali.
SOUTH EAST/S.EAST	- Comilla(part), Noakhali and Chittagong.
NORTH EAST/N.EAST	- Mymnesingh(part), Sylhet and Comilla(part).
CHT	- Chittagong Hill Tracts.

The core file of the system is an executable file named "SIM.EXE" compiled with Clipper, Summer '87 version and linked with Plink86 for compactness, although the programme is compatible with later versions of Clipper. If the corresponding sub-directory is accessible, writing "sim" and pressing "Enter" key at the "DOS" prompt of the computer will invoke the programme.

The main menu of the system has five options: "Setup", "Demand", "Supply", "Balance" and "Exit" as described below:

```
SETUP    DEMAND    SUPPLY    BALANCE    EXIT
```

A highlighting cursor will be shuttling among the menu options with the pressing of keyboard cursor keys. The use and operations of these menus are described gradually.

1. Setup

Demand, Supply and Balance sub-menus will not be activated unless "Setup" is done. It takes as input the basic information required. Pressing the "Enter" key when "Setup" is highlighted by cursor will prompt the user to enter the name of the file to use as following:

```
FILE: IP____
```

Here "IP" is the default filename and if any other name is entered, that file name will be accepted for use. After getting file name it prompts for entering the base year, number of year to simulate and the simulation year interval as following:

BASE YEAR: 1993	YEARS TO SIMULATE: 20	YEAR INTERVAL: 5

If the files associated with the entered file name is not available in the sub-directory, it creates the necessary files and all the other operations will be performed on the associated files and as per the parameters entered. Initial parameters are associated with the file system and simulation calculations. If any of the parameters is changed, a new file will be created and if any old file exists with that name will be erased. The initial parameter setup has the following flexibility:

Option	Minimum	Maximum	Default	Remarks
Data file name	-	No limit	IP	Maximum 7 characters long. with one file name, Five associated files are created and used. For example with the filename IP the associated files are: IP1.DBF, IP2.DBF, IP1.MEM, IP1.NTX, IP2.NTX
Base year	-	-	1993	Any year could be entered, but with dispersion from the default base year degree of accuracy will be reduced.
Simulation years	1	99	20	From base year.
Year interval	1	9	5	Results are calculated with this interval.

2. Demand

Demand option has sub-menus for entering initial parameters, simulation, viewing and printing the initial parameters and simulated results as shown below:

INITIATE
SIMULATE
VIEW
PRINT

There are also sub-menus for "View" and "Print" options but they are identical. They are for choosing between initial parameters and simulated results as shown below:

INITIAL PARAMETERS
SIMULATION RESULTS

Initiate - This option lets the user to enter the initial parameters for demand simulation. Once the parameters are entered they are stored in the data file and those could be edited at any time the user wishes but the results are not altered until "simulate" is not done. The parameter entry equirement for demand simulation are as follows:

Population* Of Bangladesh in the base year in thousands.
Literacy* Of Bangladesh in the base year as % of population.
Urban population* As percent of regional total for all the regions.

Population distribution
Rich population As percent of total population of Bangladesh.
Regional population As percent of total population for all regions.

Consumption in base year & consumption growth
Domestic Fuelwood Per capita consumption in m³ among urban and rural rich and poor, consumption growth rate as percent per year.
Industrial Fuelwood Per capita consumption in m³, consumption growth rate as percent per year.
Sawn timber Per capita consumption for urban rich and poor and rural population in m³, Govt. and commercial consumption as percent of urban total, consumption growth rate as percent per year, round wood equivalence factor as percent.
Post and Poles Per capita consumption in m³ and consumption growth rate as percent per year for both domestic and industrial demand.
Newsprint Per capita consumption in Kg and consumption growth rate.
Other Papers Per capita consumption in Kg and consumption growth rate.
Panel Products Consumption in m³ in all the regions.

* For other years, values are calculated from regression analysis results but user can modify them if required.

When the entry for one screen is finished a prompt will be displayed for choosing any one option from "Next", "Edit" and "Exit" at the bottom of the screen. "Next" will save the entry and bring the next entry screen, "Edit" will enable the user to edit current entries and "Exit" will exit from the initiate session without saving the current screen entries. If there is no entry requirement to follow, "Save" will be displayed in place of "Next".

Simulate - This option calculates and stores the simulated result for the simulation period with the simulation interval specified and those could be viewed or printed.

View - View option enables the user to view the initial parameters entered and projected results on the screen with such a paging facility that the user can scroll forward or backward with "PgUp" or "PgDn" keys from the keyboard. "Esc" key is used to cancel the viewing operation.

Print - Same as view, print option is used to print initial parameters and simulated results to a printer. Printer should be ready to print for 80 column printing with papers before invoking "Initial Parameters" or "Simulation Results" print options.

3. Supply

Supply sub-menu options are identical with demand options. It's sub-menus of "View" and "Print" are also identical with corresponding demand sub-options.

Initiate - Supply is calculated for Timber, Poles, Fuelwood and Pulpwood considering Natural Forest, Plantation and Village Forest as potential suppliers. For Natural Forest and Plantation initially there are six sets of entry options for each region but that number could be increased upto maximum ninety six sets to enter different types of forest and plantation data to calculate separately. There is a sub-menu to choose the region or village forest data to enter as following:

1. NORTH EAST
2. NORTH WEST
3. NORTH CENTRAL
4. WEST
5. SOUTH
6. SOUTH EAST
7. CTG HILL TRACTS
8. VILLAGE FOREST

Following are the supply parameters required for Natural Forests and Plantations in a region in base year for each set of entry option:

Productive area	Total area in hectare under production cycle for a particular type of forest or plantation.
MAI	Mean annual increase in m^3/ha/A for the particular forest or plantation.
Age	Rotation period in no. of years.
Crop density	Multiplying factor to indicate actual area under forest or plantation.
Working Cycle	Plantation period in no. of years.
Yield Rate	It is calculated by computer from other information entered in m^3/ha but user can change this value if required. Since total yield volume is calculated from this value, if only area and yield rate is entered, supply could be calculated.
Yield distribution	in percent of total yield into the categories of Timber, Pole, Fuel and Pulp.
Increase rate	Growth rate of productive area and yield in percent per year for future projection calculations.

Following are the base year supply parameters required for village forests for Timber, Pole and Fuelwood:

Volume	Supply available in m^3 for the respective region
Growth rate	In percent per year for the respective region

In case of Natural Forest or Plantation the options after finishing one screen of entries are "More", "Next", "Edit" and "Exit". "More" option will provide more six entry sets at a time.

Simulate, View and Print - These options work same way as demand options, only difference is that these options are for supply simulation, viewing and printing respectively.

4. Balance

Balance is used for on-screen display or printer output of the balance statistics from the supply and demand of forest resources for the simulation years. The sub-menu is as shown below:

VIEW
PRINT

5. Exit

Exit is used to quit from the simulation programme and return to DOS environment.

INVENTORY PROGRAMME

Introduction

Inventory programme is designed mainly for data entry, edit, store and result calculation operations of the village forest inventory carried out in six regions of Bangladesh by this project. Detail of this survey, questionnaire and calculation principles are described in the report titled "Statistical Report, Village Forest Inventory" published from this project.

This programme has limitations in this regard that, it's database, data entry, calculation and output design orientation is according to the survey questionnaire used and output requirements. But there are enough scope of expansion in this programme to use for future village forest surveys or further analysis and manipulation of the current data.

Programme tree structure, programme code and data file structures are included in Appendix 4.

Layout and Operation

Initially the programme modules were scattered as described in the village forest inventory report but now they are integrated into three programme files with one calling programme file named "VFI.PRG". Others are the procedure files named "VFIEN.PRG" and "VFIPR.PRG" for entry and printing operations respectively. To operate the programme, it should be invoked with Dbase by typing "dbase vfi" at the dos command prompt of the corresponding sub-directory where the programme, data and format files are kept and the dbase programme is in the path list. It could also be invoked by typing "do vfi" at the dbase dot command prompt.

The opening menu of the programme has three options "Entry", "Print" and "Exit" as shown below:

```
1 - ENTRY
2 - PRINT
3 - EXIT
```

Respective operation could be invoked by pressing corresponding number at the choice prompt.

1. Entry

Entry option is used for data entry edit or view the data files. For convenience the entry option is divided into two groups. The options will be displayed as below:

```
1 - PART I
2 - PART II & III

Enter Choice:
```

Pressing 1 or 2 will take the user to the next menu for edit as below:

```
1 > DATA ENTRY
2 > VIEW/EDIT
Q > QUIT

MAKE YOUR CHOICE >
```

Pressing 1 will take the user to the data entry options in formatted screens sequentially. Choosing option 2 will display the following option screen for Part I entry:

```
1 > GENERAL INFORMATION
2 > BAMBOOS
3 > THIN TREES
4 > LARGE TREES
5 > PALM TREES
6 > NO WOOD VALUE
Q > QUIT

MAKE YOUR CHOICE >
```

Entering corresponding number at the choice prompt will provide browsing facility of the data files used to store respective data. For the viewing and editing of Part II and Part III will display the following options:

```
1 > FUEL AND FODDER
2 > BUILDING MATERIAL
3 > FURNITURE
4 > AGRICULTURAL IMPLEMENTS
5 > TRANSPORTATION
6 > SALES & PURCHASE
7 > HARVEST
8 > ATTITUDE
Q > QUIT

MAKE YOUR CHOICE >
```

In every level "Quit" is for exiting to the previous level.

2. Print

Print is used for calculating the results and printing them in a simple format according to the master plan requirement. Following are the print options available from the programme:

```
 1: LIVESTOCK NUMBER
 2: STOCK OF PALM TREES: ALL STRATA
 3: STOCK VOLUME AND STAND TABLE
 4: STOCK OF TREES OF NO WOOD VALUE
 5: CANES: NO. OF CULMS
 6: FUEL COLLECTIONS
 7: TREE STOCK: DIA <=8" & HT.>5'
 8: BAMBOO RESOURCES
 9: DISTANCE TO GO TO COLLECT FUEL
10: RANKING OF AVAILABILITY OF FUEL
11: TREE STOCK - NO WOOD VALUE: ALL STRATA
12: BAMBOO RESOURCES - ACCORDING TO LANDOWNERSHIP
13: THATCH GRASS AREA
14: FUEL CONSUMPTION: ALL STRATA
15: NO. OF DAYS IN A WEEK USED IN FUEL COLLECTION
16: DISTANCE TO TRAVERSE TO COLLECT FODDER
17: RANKING OF AVAILABILITY OF FODDER
18: WOOD AND BAMBOO USE - IN TRANSPORTATION
```

19: WOOD AND BAMBOO USE - IN BUILDING AND FENCING
20: SALES AND PURCHASE OF TIMBER AND BAMBOO
21: POPULATION, OCCUPATION AND LANDOWNERSHIP
22: BAMBOO RESOURCES - CLUMP
23: STOCK OF PALM TREES
24: TREE STOCK: DIA <=8" & HT>5' - BY LANDOWNERSHIP
25: STOCK OF TREES: 4" < DIA < 8"
26: POPULATION DISTRIBUTION
27: STOCK VOL. BY LANDHOLDING

Prompts at the bottom of the print option are as follows:

```
Enter Number to Print --------->
Send Output to Printer(y/n) -->
```

First option is to choose the item to print and second option is for output routing. Pressing "y" at the output option will route the result to the printer, otherwise it will be displayed on the screen. Since this programme is linear in its construction, every time a print request is made the programme will need to go through the time consuming operation of reading and calculation of loaded database files. If this programme is developed in future, the print routines could be divided into calculation and print routines, so that all the calculations and reading associated with a data file could be done in a single run and results dumped into an intermediate result data file. This will require plenty of processing time and the print routines will need to use the result data files only.

File System

The village survey questionnaire consists of three parts. Part I contains general information, and supply information on distinct forest products which were listed separately. Similarly, Part II contains consumption information on distinct items, shown separately. For efficient and effective file maintenance, a file was created for each distinct item shown separately in Part I and Part II. All information in Part III were compiled in a single file. Details of the questionnaire is included in the report titled "Village Inventory and Household Consumption Report" published by this project. Associated files with this programme and their contents are listed below:

File Name	Content
Programme files	
VFI.PRG	Main calling programme
VFIEN.PRG	Procedure modules for data entry and edit
VFIPR.PRG	Procedure modules for printing and other operation
Data files	
General	
PDATA.DBF	Population and other data necessory for calculation
Part I	
PART101.DBF	General information, Cane (supply) and Thatch grass (supply)
PART102.DBF	Bamboo (supply)
PART103.DBF	Regenerating Trees (supply)
PART104.DBF	Trees (supply)
PART105.DBF	Palm Trees (supply)
PART106.DBF	Trees which have no wood value (supply)

Part II:

PART201.DBF	Fuel (consumption)
PART202.DBF	Building Material (consumption)
PART203.DBF	Furniture (consumption)
PART204.DBF	Agricultural Implements
PART205.DBF	Transportation
PART206.DBF	Sales and Purchase
PART207.DBF	Harvest

Part III:

PART301.DBF	Public Attitude

Index files

ID101.NDX	Index file of PART101.DBF on field ID
CODE103.NDX	Index file of PART103.DBF on field CODE
CODE104.NDX	Index file of PART104.DBF on field CODE
PDATA.NDX	Index file of PDATA.DBF on field STRTA

Entry format files

Part I:

PART101.FMT	General information, Cane and Thatch grass data
PART102.FMT	Bamboo data
PART103.FMT	Regenerating Trees
PART104.FMT	Trees
PART105.FMT	Palm Trees
PART106.FMT	Trees which have no wood value

Part II:

PART201.FMT	Fuel
PART202.FMT	Building Material
PART203.FMT	Furniture
PART204.FMT	Agricultural Implements
PART205.FMT	Transportation
PART206.FMT	Sales and Purchase
PART207.FMT	Harvest

Part III:

PART301.FMT	Public Attitude

APPENDIX 1
ABBREVIATIONS AND TERMS

PROJECT 372001/25
FORESTRY MASTER PLAN
BANGLADESH TA 1355-BAN

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 05 NOVEMBER 1992

SYSTEMS ANALYSIS

APPENDIX 1
ABBREVIATIONS AND TERMS

ABBREVIATIONS

ACF	- Assistant Conservator of Forests
ADB	- Asian Development Bank
BFD	- Bangladesh Forest Department
CHT	- Chittagong Hill Tracts
CTG	- Chittagong
Dbase, dBase	- Data Base
DBH	- Diameter at Breast Height
Dia	- Diameter
DLU	- Discrete Land Unit
DOS	- Disk Operating System
FAO	- Food and Agriculture Organization of the United Nations
FD	- Forest Department
GIS	- Geographic Information System
gm	- Gram
GOB	- Government of Bangladesh
ha	- Hectare
Ht	- Height
IBM	- International Business Machines
kg	- Kilogram
m	- Metre
m^3/ha	- Cubic metre per hectare
m^3	- Cubic metre
m^3/ha/A	- Cubic metre per hectare per annum
MAI	- Mean annual increment
Mbyte	- Mega Byte
MHz	- Mega Hertz
No.	- Number
RAM	- Random Access Memory
RIMS	- Resource Information Management System
RMS	- Resource Management System
Tk	- Taka
UNDP	- United Nations Development Programme
VGA	- Video Graphic Array
WB	- World Bank

TERMS

Apple IIe	- Computer system named Apple Macintosh, IIe is model number.
AppleWork	- Spreadsheet programme used in Apple Macintosh computer.
Clipper, Summer '87	- Name of a database programme compiler. Summer '87 is for version.
Compiler	- Converts high level programme codes into executable instruction sets.

Cursor - Blip or highlighted point on the screen showing working position.

Dbase, Dbase III - Dbase is a computer data base package programme, III its version number.

Default - What is originally assumed and set in the computer programme but may be altered.

Digitizer - One type of pointing device to precisely point on the computer screen corresponding to the points on a two dimentional printed map or drawing.

DOS - Computer operating system at the base level.

Enter - Computer key usually used to instruct computer to start any operation.

Esc - Computer key usually used to cancel or abort from any operation.

Executable file - Programme file containing instructions to operate on data and computer.

FoxPro - Name of a computer database package programme.

GIS - Computer package based on utilization of geographic map data. It is used to draw maps, store map data, incorporating data and attributes with maps, manipulating maps, displaying and printing maps with user defined colours, symbols, shades etc.

Hard Disk - Computer permanent data storage unit.

IBM compatible - IBM Computer is a computer system known as personal computer or PC. Clone computer equivalent to the original IBM PC is called IBM compatible computer.

Mbyte (Mega Byte) - Million bytes, Byte is a small size unit for computer data.

Memory - Computer temporary data storage space used to store programme instructions and data files. RAM indicates the physical memory units.

MHz (Mega Hertz) - Million cycles per second, to designate computer processing speed in terms of main processor speed.

Mouse - One type of computer screen pointing and instruction device.

PgUp, PgDn - Computer key for paging upward and downward.

Plink86 - Name of a computer programme to link executable instruction sets.

Plotter - Printing device used to get larger and accurate print from the computer, essential for map or drawing printing.

Programme code - Instruction sets understandable by computer programmes.

Sub-directory - Smaller sub-division of computer hard disk storage space, separated purposefully and designated with a name for storage of computer files.

Super VGA, SVGA - Indicates high resolution display facility of the computer monitor.

APPENDIX 2
TERMS OF REFERENCE

PROJECT 372001/25
FORESTRY MASTER PLAN
BANGLADESH TA 1355-BAN

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 05 NOVEMBER 1992

SYSTEMS ANALYSIS

APPENDIX 2
TERMS OF REFERENCE

1. Review the Resource Information Management System installed recently at the Forest Depertment and suggest any measures necessary to optimize the utilization of the facilities now in place.

2. Prepare a computer programme to collate and assess forest products supply and demand data which can be updated at regular intervals.

APPENDIX 3
SIMULATION PROGRAMME STRUCTURE AND CODE

PROJECT 372001/25
FORESTRY MASTER PLAN
BANGLADESH TA 1355-BAN

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 05 NOVEMBER 1992

SYSTEMS ANALYSIS

APPENDIX 3
SIMULATION PROGRAMME STRUCTURE AND CODE

TABLE OF CONTENTS

1. TREE DIAGRAM

```
SIM.PRG
├────RT_ARROW (procedure)
├────LT_ARROW (procedure)
├────BY_INI (procedure)
│   │   &F_NAME1 (database)
│   │   &F_NAME2 (database)
│   │   &F_NAME1 (index file)
│   │   &F_NAME2 (index file)
│   │   &F_NAME1 (memory file)
│   └────CR_IP2() (function)
│        │  TEMP1.DBF (database)
│        │  &MF_NAME1 (database)
│        │  TEMP.DBF (database)
│        │  &MF_NAME2 (database)
│        │  &MF_NAME1 (index file)
│        │  &MF_NAME2 (index file)
│        └────FLD_I (procedure)
├────DEMAND (procedure)
│   ├────D_INI (procedure)
│   │    │  &F_NAME2 (database)
│   │    │  &F_NAME1 (database)
│   │    │  &F_NAME2 (index file)
│   │    │  &F_NAME1 (index file)
│   │    │  &F_NAME1 (memory file)
│   │    ├────INI_2 (procedure)
│   │    ├────DB_2 (procedure)
│   │    ├────INI_1 (procedure)
│   │    ├────INI_3 (procedure)
│   │    ├────DB_1 (procedure)
│   │    └────DB_3 (procedure)
│   ├────D_SIM (procedure)
│   │       &F_NAME2 (database)
│   │       &F_NAME1 (database)
│   │       &F_NAME2 (index file)
│   │       &F_NAME1 (index file)
│   ├────D_VW (procedure)
│   │    ├────Z_NOTE (procedure)
│   │    ├────D_VW1 (procedure)
│   │    │    │  &F_NAME2 (database)
│   │    │    │  &F_NAME1 (database)
│   │    │    │  &F_NAME2 (index file)
│   │    │    │  &F_NAME1 (index file)
│   │    │    ├────INI_2 (procedure)
│   │    │    └────&PROX
│   │    └────D_VW2 (procedure)
│   │         │  &F_NAME2 (database)
│   │         │  &F_NAME2 (index file)
│   │         └────&PROX (procedure)
│   ├────D_PR (procedure)
│   │    ├────D_PR1(procedure)
│   │    │    │  &F_NAME2 (database)
│   │    │    │  &F_NAME1 (database)
│   │    │    │  &F_NAME2 (index file)
│   │    │    │  &F_NAME1 (index file)
│   │    │    ├────INI_1 (procedure)
│   │    │    ├────INI_3 (procedure)
│   │    │    └────INI_2 (procedure)
│   │    └────D_PR2 (procedure)
│   │         │  &F_NAME2 (database)
│   │         │  &F_NAME2 (index file)
│   │         ├────D_PR2X (procedure)
│   │         │    └────I_TOT (procedure)
│   │         └────D_PR2Y (procedure)
│   │              └────I_TOT (procedure)
│   ├────RT_ARROW (procedure)
│   └────LT_ARROW (procedure)
├────SUPPLY (procedure)
│   ├────CIRCLE (procedure)
│   │    ├────S_INI (procedure)
│   │    │    │  &F_NAME1 (database)
│   │    │    │  &F_NAME1 (index file)
│   │    │    │  &F_NAME1 (memory file)
│   │    │    ├────INI_5 (procedure)
│   │    │    └────DB_5 (procedure)
│   │    └────VS_INI (procedure)
│   │         │  &F_NAME1 (database)
│   │         │  &F_NAME1 (index file)
│   │         ├────INI_3 (procedure)
│   │         └────DB_3 (procedure)
│   ├────S_SIM (procedure)
│   │    │  &F_NAME2 (database)
│   │    │  &F_NAME1 (database)
│   │    │  &F_NAME2 (index file)
│   │    │  &F_NAME1 (index file)
│   │    ├────INI_5 (procedure)
│   │    └────INI_3 (procedure)
│   ├────S_VW (procedure)
│   │    ├────Z_NOTE (procedure)
│   │    ├────S_VW1 (procedure)
│   │    │    │  &F_NAME1 (database)
│   │    │    │  &F_NAME1 (index file)
│   │    │    ├────INI_3 (procedure)
│   │    │    ├────S_NF (procedure)
│   │    │    │    └────INI_5 (procedure)
│   │    │    ├────S_PL (procedure)
│   │    │    │    └────INI_5 (procedure)
│   │    │    └────SPAGE15 (procedure)
│   │    └────S_VW2 (procedure)
│   │         │  &F_NAME2 (database)
│   │         │  &F_NAME2 (index file)
│   │         └────&PROX (procedure)
│   ├────S_PR (procedure)
│   │    ├────S_PR1 (procedure)
│   │    │    │  &F_NAME1 (database)
│   │    │    │  &F_NAME1 (index file)
│   │    │    ├────INI_3 (procedure)
│   │    │    └────INI_5 (procedure)
│   │    └────S_PR2 (procedure)
│   │         │  &F_NAME2 (database)
│   │         │  &F_NAME2 (index file)
│   │         └────S_PR2X (procedure)
│   │              └────I_TOT (procedure)
│   ├────RT_ARROW (procedure)
│   └────LT_ARROW (procedure)
└────BALANCE (procedure)
     ├────B_VW (procedure)
     │    │  &F_NAME2 (database)
     │    │  &F_NAME2 (index file)
     │    └────&PROX (procedure)
     ├────B_PR (procedure)
     │    │  &F_NAME2 (database)
     │    │  &F_NAME2 (index file)
     │    ├────I_T8 (procedure)
     │    └────I_TY (procedure)
     ├────RT_ARROW (procedure)
     └────LT_ARROW (procedure)
```

2. PROGRAMME CODE

```
*:***************************************
*:   Program: SIM.PRG
*:
*:   Procedures and  Functions
*:
*:                  : RT_ARROW
*:                  : LT_ARROW
*:                  : BY_INI
*:                  : DEMAND
*:                  : SUPPLY
*:                  : BALANCE
*:                  : CIRCLE
*:                  : FLD_I
*:                  : CR_IP2()
*:                  : D_INI
*:                  : INI_1
*:                  : DB_1
*:                  : INI_2
*:                  : DB_2
*:                  : INI_3
*:                  : DB_3
*:                  : INI_4
*:                  : DB_4
*:                  : D_SIM
*:                  : D_VW
*:                  : D_VW1
*:                  : PAGE1
*:                  : PAGE2
*:                  : PAGE3
*:                  : D_VW2
*:                  : I_TOT
*:                  : PGS1
*:                  : D_V_R1
*:                  : PGS2
*:                  : PGS3
*:                  : PGS4
*:                  : PGS5
*:                  : PGS6
*:                  : D_V_R2
*:                  : PGS7
*:                  : PGS8
*:                  : INI_5
*:                  : DB_5
*:                  : S_INI
*:                  : VS_INI
*:                  : S_SIM
*:                  : S_VW
*:                  : S_VW1
*:                  : S_NF
*:                  : S_PL
*:                  : SPAGE15
*:                  : S_VW2
*:                  : SPGS1
*:                  : S_V_R1
*:                  : SPGS2
*:                  : SPGS3
*:                  : SPGS4
*:                  : SPGS5
*:                  : SPGS6
*:                  : SPGS7
*:                  : SPGS8
*:                  : SPGS9
*:                  : SPGS10
*:                  : SPGS11
*:                  : Z_NOTE
*:                  : B_VW
*:                  : I_T8
*:                  : I_TY
*:                  : BPGS1
*:                  : BPGS2
*:                  : BPGS3
*:                  : BPGS4
*:                  : BPGS5
*:                  : D_PR
*:                  : D_PR1
*:                  : D_PR2X
*:                  : D_PR2Y
*:                  : D_PR2
*:                  : S_PR
*:                  : S_PR1
*:                  : S_PR2X
*:                  : S_PR2
*:                  : B_PR
*:
*:     Calls
*:
*:     : RT_ARROW (procedure in SIM.PRG)
*:     : LT_ARROW (procedure in SIM.PRG)
*:     : BY_INI     (procedure in SIM.PRG)
*:     : DEMAND    (procedure in SIM.PRG)
*:     : SUPPLY          (procedure in SIM.PRG)
*:     : BALANCE   (procedure in SIM.PRG)
*:***************************************
SET WRAP ON
SET EXACT OFF
SET DATE ITALIAN
IF ISCOLOR()
      CLR1="W+/B"
      CLR2="BG+/B,GR+/R"
      CLR3="GR+/GR,BG+/N"
ELSE
      CLR1="W/N,N/W"
      CLR2="W/N,N/W"
      CLR3="W/N,N/W"
ENDIF
SET COLOR TO &CLR1
CLEAR
@1,0 TO 1,79 DOUBLE
@2,0 TO 2,79 DOUBLE
SET COLOR TO &CLR2
@4,0 SAY  REPL("▓",80)
@1,27 SAY "[ FORESTRY MASTER PLAN ]"
@2,23 SAY "[SUPPLY AND DEMAND SIMULATION]"
DECLARE M[5]
M[1]="   SETUP    "
M[2]="  DEMAND   "
M[3]="  SUPPLY   "
M[4]="  BALANCE  "
M[5]="    EXIT   "
DECLARE D[4]
D[1]=" INITIATE "
D[2]=" SIMULATE "
D[3]=" VIEW      "
D[4]=" PRINT     "
DECLARE C[8]
C[1]=" 1. NORTH EAST       "
C[2]=" 2. NORTH WEST       "
C[3]=" 3. NORTH CENTRAL    "
C[4]=" 4. WEST             "
C[5]=" 5. SOUTH            "
C[6]=" 6. SOUTH EAST       "
C[7]=" 7. CTG HILL TRACTS  "
C[8]=" 8. VILLAGE FOREST   "
DECLARE Z_STR[7]
Z_STR[1]="N_WEST"
Z_STR[2]="N_CENT"
Z_STR[3]="WEST"
Z_STR[4]="SOUTH"
```

```
Z_STR[5]="S_EAST"
Z_STR[6]="N_EAST"
Z_STR[7]="CHT"
SET COLOR TO &CLR2
M1=1
PUBLIC Y_B,Y_I,Y_N,Y_R,Y_DI,Y_NF
PUBLIC Y_PL,SCR1,SCR2,F_ST,F_NAME1
PUBLIC F_NAME2,DF_I1,F_NAME

DF_I1="N"
Y_DI="N"
DO WHILE .T.
    @3,0 TO 5,14 DOUBLE
    @3,17 TO 5,30 DOUBLE
    @3,33 TO 5,46 DOUBLE
    @3,49 TO 5,63 DOUBLE
    @3,66 TO 5,79 DOUBLE
    @4,1 PROMPT M[1]
    @4,18 PROMPT M[2]
    @4,34 PROMPT M[3]
    @4,50 PROMPT M[4]
    @4,67 PROMPT M[5]
    MENU TO M1
    SET COLOR TO &CLR3
    SAVE SCREEN TO SCR1
    SET KEY 4 TO RT_ARROW
    SET KEY 19 TO LT_ARROW
    DO CASE
    CASE M1=1
        DO BY_INI
    CASE M1=2.AND.DF_I1="Y"
        D1=1
        DO DEMAND
    CASE M1=3.AND.DF_I1="Y"
        S1=1
        DO SUPPLY
    CASE M1=4.AND.DF_I1="Y"
        B1=1
        DO BALANCE
    CASE M1=5
        !CLS
        QUIT
    ENDCASE
    SET KEY 4 TO
    SET KEY 19 TO
    RESTO SCREEN FROM SCR1
    SET COLOR TO &CLR2
ENDDO
RETURN

*!*************************************
*!    Procedure: DEMAND
*!    Called by: SIM.PRG
*!    Calls
*!    : D_INI       (procedure in SIM.PRG)
*!    : D_SIM       (procedure in SIM.PRG)
*!    : D_VW        (procedure in SIM.PRG)
*!    : D_PR        (procedure in SIM.PRG)
*!    : RT_ARROW (procedure in SIM.PRG)
*!    : LT_ARROW (procedure in SIM.PRG)
*!*************************************
PROC DEMAND
DO WHILE D1#0
    @6,17 TO 11,30 DOUBLE
    @7,18 PROMPT D[1]
    @8,18 PROMPT D[2]
    @9,18 PROMPT D[3]
    @10,18 PROMPT D[4]
    MENU TO D1
    IF D1#0
        SET KEY 4 TO
        SET KEY 19 TO
        SAVE SCREEN TO SCR2
        DO CASE
        CASE D1=1
            DO D_INI
        CASE D1=2
            DO D_SIM
        CASE D1=3
            D2=1
            DO D_VW
        CASE D1=4
            P2=1
            DO D_PR
        ENDCASE
        CLOSE DATABASES
        REST SCREEN FROM SCR2
        SET KEY 4 TO RT_ARROW
        SET KEY 19 TO LT_ARROW
    ENDIF
ENDDO
RETURN

*!*************************************
*!    Procedure: SUPPLY
*!    Called by: SIM.PRG
*!    Calls:
*!    : CIRCLE      (procedure in SIM.PRG)
*!    : S_SIM   (procedure in SIM.PRG)
*!    : S_VW        (procedure in SIM.PRG)
*!    : S_PR        (procedure in SIM.PRG)
*!    : RT_ARROW            (procedure in SIM.PRG)
*!    : LT_ARROW            (procedure in SIM.PRG)
*!*************************************
PROC SUPPLY
DO WHILE S1# 0
    @6,33 TO 11,46 DOUBLE
    @7,34 PROMPT D[1]
    @8,34 PROMPT D[2]
    @9,34 PROMPT D[3]
    @10,34 PROMPT D[4]
    MENU TO S1
    IF S1# 0
        SET KEY 4 TO
        SET KEY 19 TO
        SAVE SCREEN TO SCR2
        DO CASE
        CASE S1=1
            C1=1
            DO CIRCLE
        CASE S1=2
            DO S_SIM
        CASE S1=3
            S2=1
            DO S_VW
        CASE S1=4
            P2=1
            DO S_PR
        ENDCASE
        CLOSE DATABASES
        REST SCREEN FROM SCR2
        SET KEY 4 TO RT_ARROW
        SET KEY 19 TO LT_ARROW
    ENDIF
ENDDO
RETURN
*!*************************************
*!    Procedure: BALANCE
*!    Called by: SIM.PRG
*!    Calls
*!    : B_VW
*!    : B_PR              (procedure in SIM.PRG)
                              (procedure in SIM.PRG)
```

```
*!     : RT_ARROW (procedure in SIM.PRG)
*!     : LT_ARROW (procedure in SIM.PRG)
*!**************************************
PROC BALANCE
DO WHILE B1#0
     @6,49 TO 9,63 DOUBLE
     @7,50 PROMPT " "+D[3]
     @8,50 PROMPT " "+D[4]
     MENU TO B1
     IF B1#0
          SET KEY 4 TO
          SET KEY 19 TO
          SAVE SCREEN TO SCR2
          DO CASE
          CASE B1=1
               DO B_VW
          CASE B1=2
               DO B_PR
          ENDCASE
          CLOSE DATABASES
          REST SCREEN FROM SCR2
          SET KEY 4 TO RT_ARROW
          SET KEY 19 TO LT_ARROW
     ENDIF
ENDDO
RETURN

*!**************************************
*!        Procedure: RT_ARROW
*!        Called by: SIM.PRG
*!        : DEMAND          (procedure in SIM.PRG)
*!        : SUPPLY          (procedure in SIM.PRG)
*!        : BALANCE         (procedure in SIM.PRG)
*!**************************************
PROC RT_ARROW
IF M1=2.OR.M1=3
    KEYBOARD CHR(27)+CHR(4)+CHR(13)
ELSE
    KEYBOARD CHR(27)+CHR(4)
ENDIF
RETURN
*!**************************************
*!        Procedure: LT_ARROW
*!        Called by: SIM.PRG
*!        : DEMAND          (procedure in SIM.PRG)
*!        : SUPPLY          (procedure in SIM.PRG)
*!        : BALANCE         (procedure in SIM.PRG)
*!**************************************
PROC LT_ARROW
IF M1=3.OR.M1=4
    KEYBOARD CHR(27)+CHR(19)+CHR(13)
ELSE
    KEYBOARD CHR(27)+CHR(19)
ENDIF
RETURN
*!**************************************
*!    Procedure: CIRCLE
*!    Called by: SUPPLY              (proc in SIM.PRG)
*!    Calls
*!          : S_INI             (procedure in SIM.PRG)
*!          : VS_INI            (procedure in SIM.PRG)
*!**************************************
PROC CIRCLE
SAVE SCREEN TO SCR2
DO WHILE C1#0
     @12,28 TO 21,51 DOUBLE
     @13,29 PROMPT C[1]
     @14,29 PROMPT C[2]
     @15,29 PROMPT C[3]
     @16,29 PROMPT C[4]
     @17,29 PROMPT C[5]
     @18,29 PROMPT C[6]
     @19,29 PROMPT C[7]
     @20,29 PROMPT C[8]
     MENU TO C1
     IF C1#0
          IF C1#8
               DO S_INI
          ELSE
               DO VS_INI
          ENDIF
          RESTO SCREEN FROM SCR2
     ENDIF
ENDDO
RETURN
*!**************************************
*!    Procedure: FLD_I
*!    Called by: CR_IP2()            (func in SIM.PRG)
*!**************************************
PROC FLD_I
REPLACE FIELD_TYPE WITH "N"
REPLACE FIELD_LEN WITH 14
REPLACE FIELD_DEC WITH 5
RETURN
*!**************************************
*!    Function: CR_IP2()
*!    Called by: BY_INI              (proc in SIM.PRG)
*!    Calls
*!    : FLD_I   (procedure in SIM.PRG)
*!    Uses
*!          : TEMP1.DBF
*!          : &MF_NAME1
*!          : TEMP.DBF
*!          : &MF_NAME2
*!    Indexes
*!          : &MF_NAME1
*!          : &MF_NAME2
*!**************************************
FUNCTION CR_IP2
PARAMETERS MF_NAME1, MF_NAME2
DECLARE I_LST[12]
I_LST[1]="G_RATE1"
I_LST[2]="N_WEST"
I_LST[3]="N_CENT"
I_LST[4]="WEST"
I_LST[5]="SOUTH"
I_LST[6]="S_EAST"
I_LST[7]="N_EAST"
I_LST[8]="CHT"
I_LST[9]="U_RICH"
I_LST[10]="U_POOR"
I_LST[11]="R_RICH"
I_LST[12]="R_POOR"
SELE A
CREATE TEMP1
USE TEMP1
APPEND BLANK
REPLACE FIELD_NAME WITH "ITEM1"
REPLACE FIELD_TYPE WITH "C"
REPLACE FIELD_LEN WITH 20
FOR X=1 TO 12
       APPEND BLANK
       REPL FIELD_NAME WITH I_LST[X]
       DO FLD_I
NEXT
USE
CREATE &MF_NAME1 FROM TEMP1
ERASE TEMP1.DBF
USE &MF_NAME1
INDEX ON ITEM1 TO &MF_NAME1
DECLARE I_RC[12]
I_RC[1]="FULWD_DU"
```

```
I_RC[2]="FULWD_IU"
I_RC[3]="PT/PL_DU"
I_RC[4]="PT/PL_IU"
I_RC[5]="STMBR"
I_RC[6]="NWPT"
I_RC[7]="PAPR"
I_RC[8]="PANEL"
FOR X=1 TO 8
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="NF1_AREA"
I_RC[2]="NF1_MAI"
I_RC[3]="NF1_AGE"
I_RC[4]="NF1_CRD"
I_RC[5]="NF1_W_C"
I_RC[6]="NF1_YLD"
I_RC[7]="NF1_TMBR"
I_RC[8]="NF1_POLE"
I_RC[9]="NF1_FUEL"
I_RC[10]="NF1_PULP"
I_RC[11]="NF1_GR_A"
I_RC[12]="NF1_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="NF2_AREA"
I_RC[2]="NF2_MAI"
I_RC[3]="NF2_AGE"
I_RC[4]="NF2_CRD"
I_RC[5]="NF2_W_C"
I_RC[6]="NF2_YLD"
I_RC[7]="NF2_TMBR"
I_RC[8]="NF2_POLE"
I_RC[9]="NF2_FUEL"
I_RC[10]="NF2_PULP"
I_RC[11]="NF2_GR_A"
I_RC[12]="NF2_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="NF3_AREA"
I_RC[2]="NF3_MAI"
I_RC[3]="NF3_AGE"
I_RC[4]="NF3_CRD"
I_RC[5]="NF3_W_C"
I_RC[6]="NF3_YLD"
I_RC[7]="NF3_TMBR"
I_RC[8]="NF3_POLE"
I_RC[9]="NF3_FUEL"
I_RC[10]="NF3_PULP"
I_RC[11]="NF3_GR_A"
I_RC[12]="NF3_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="NF4_AREA"
I_RC[2]="NF4_MAI"
I_RC[3]="NF4_AGE"
I_RC[4]="NF4_CRD"
I_RC[5]="NF4_W_C"
I_RC[6]="NF4_YLD"
I_RC[7]="NF4_TMBR"
I_RC[8]="NF4_POLE"
I_RC[9]="NF4_FUEL"
I_RC[10]="NF4_PULP"
I_RC[11]="NF4_GR_A"
I_RC[12]="NF4_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="NF5_AREA"
I_RC[2]="NF5_MAI"
I_RC[3]="NF5_AGE"
I_RC[4]="NF5_CRD"
I_RC[5]="NF5_W_C"
I_RC[6]="NF5_YLD"
I_RC[7]="NF5_TMBR"
I_RC[8]="NF5_POLE"
I_RC[9]="NF5_FUEL"
I_RC[10]="NF5_PULP"
I_RC[11]="NF5_GR_A"
I_RC[12]="NF5_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="NF6_AREA"
I_RC[2]="NF6_MAI"
I_RC[3]="NF6_AGE"
I_RC[4]="NF6_CRD"
I_RC[5]="NF6_W_C"
I_RC[6]="NF6_YLD"
I_RC[7]="NF6_TMBR"
I_RC[8]="NF6_POLE"
I_RC[9]="NF6_FUEL"
I_RC[10]="NF6_PULP"
I_RC[11]="NF6_GR_A"
I_RC[12]="NF6_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="PL1_AREA"
I_RC[2]="PL1_MAI"
I_RC[3]="PL1_AGE"
I_RC[4]="PL1_CRD"
I_RC[5]="PL1_W_C"
I_RC[6]="PL1_YLD"
I_RC[7]="PL1_TMBR"
I_RC[8]="PL1_POLE"
I_RC[9]="PL1_FUEL"
I_RC[10]="PL1_PULP"
I_RC[11]="PL1_GR_A"
I_RC[12]="PL1_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="PL2_AREA"
I_RC[2]="PL2_MAI"
I_RC[3]="PL2_AGE"
I_RC[4]="PL2_CRD"
I_RC[5]="PL2_W_C"
I_RC[6]="PL2_YLD"
I_RC[7]="PL2_TMBR"
I_RC[8]="PL2_POLE"
I_RC[9]="PL2_FUEL"
I_RC[10]="PL2_PULP"
I_RC[11]="PL2_GR_A"
I_RC[12]="PL2_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="PL3_AREA"
I_RC[2]="PL3_MAI"
I_RC[3]="PL3_AGE"
```

```
I_RC[4]="PL3_CRD"
I_RC[5]="PL3_W_C"
I_RC[6]="PL3_YLD"
I_RC[7]="PL3_TMBR"
I_RC[8]="PL3_POLE"
I_RC[9]="PL3_FUEL"
I_RC[10]="PL3_PULP"
I_RC[11]="PL3_GR_A"
I_RC[12]="PL3_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="PL4_AREA"
I_RC[2]="PL4_MAI"
I_RC[3]="PL4_AGE"
I_RC[4]="PL4_CRD"
I_RC[5]="PL4_W_C"
I_RC[6]="PL4_YLD"
I_RC[7]="PL4_TMBR"
I_RC[8]="PL4_POLE"
I_RC[9]="PL4_FUEL"
I_RC[10]="PL4_PULP"
I_RC[11]="PL4_GR_A"
I_RC[12]="PL4_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="PL5_AREA"
I_RC[2]="PL5_MAI"
I_RC[3]="PL5_AGE"
I_RC[4]="PL5_CRD"
I_RC[5]="PL5_W_C"
I_RC[6]="PL5_YLD"
I_RC[7]="PL5_TMBR"
I_RC[8]="PL5_POLE"
I_RC[9]="PL5_FUEL"
I_RC[10]="PL5_PULP"
I_RC[11]="PL5_GR_A"
I_RC[12]="PL5_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="PL6_AREA"
I_RC[2]="PL6_MAI"
I_RC[3]="PL6_AGE"
I_RC[4]="PL6_CRD"
I_RC[5]="PL6_W_C"
I_RC[6]="PL6_YLD"
I_RC[7]="PL6_TMBR"
I_RC[8]="PL6_POLE"
I_RC[9]="PL6_FUEL"
I_RC[10]="PL6_PULP"
I_RC[11]="PL6_GR_A"
I_RC[12]="PL6_GR_Y"
FOR X=1 TO 12
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
I_RC[1]="VF_TMBR"
I_RC[2]="VF_POLE"
I_RC[3]="VF_FUEL"
I_RC[4]="VF_GR_T"
I_RC[5]="VF_GR_P"
I_RC[6]="VF_GR_F"
FOR X=1 TO 6
     APPEND BLANK
     REPL ITEM1 WITH I_RC[X]
NEXT
```

```
USE
CREATE TEMP
USE TEMP
APPEND BLANK
REPLACE FIELD_NAME WITH "ITEM2"
REPLACE FIELD_TYPE WITH "C"
REPLACE FIELD_LEN WITH 20
APPEND BLANK
REPLACE FIELD_NAME WITH "G_RATE2"
DO FLD_I
FOR I=Y_B TO Y_B+Y_N STEP Y_I
     M_FNAME="Y"+STR(I,4,0)
     APPEND BLANK
     REPLACE FIELD_NAME WITH M_FNAME
     DO FLD_I
NEXT
USE
CREATE &MF_NAME2 FROM TEMP
ERASE TEMP.DBF
USE &MF_NAME2
INDEX ON ITEM2 TO &MF_NAME2
DECLARE I_RC[35]
I_RC[1]="T_POP"
I_RC[2]="U_POP_1"
I_RC[3]="U_POP_2"
I_RC[4]="U_POP_3"
I_RC[5]="U_POP_4"
I_RC[6]="U_POP_5"
I_RC[7]="U_POP_6"
I_RC[8]="U_POP_7"
I_RC[9]="NW_POP_1"
I_RC[10]="NW_POP_2"
I_RC[11]="NW_POP_3"
I_RC[12]="NW_POP_4"
I_RC[13]="NC_POP_1"
I_RC[14]="NC_POP_2"
I_RC[15]="NC_POP_3"
I_RC[16]="NC_POP_4"
I_RC[17]="W_POP_1"
I_RC[18]="W_POP_2"
I_RC[19]="W_POP_3"
I_RC[20]="W_POP_4"
I_RC[21]="S_POP_1"
I_RC[22]="S_POP_2"
I_RC[23]="S_POP_3"
I_RC[24]="S_POP_4"
I_RC[25]="SE_POP_1"
I_RC[26]="SE_POP_2"
I_RC[27]="SE_POP_3"
I_RC[28]="SE_POP_4"
I_RC[29]="NE_POP_1"
I_RC[30]="NE_POP_2"
I_RC[31]="NE_POP_3"
I_RC[32]="NE_POP_4"
I_RC[33]="CHT_POP_1"
I_RC[34]="CHT_POP_2"
I_RC[35]="CHT_POP_3"
FOR X=1 TO 35
     APPEND BLANK
     REPL ITEM2 WITH I_RC[X]
NEXT
I_RC[1]="CHT_POP_4"
I_RC[2]="LITERACY"
I_RC[3]="NW_FWD_D"
I_RC[4]="NC_FWD_D"
I_RC[5]="W_FWD_D"
I_RC[6]="S_FWD_D"
I_RC[7]="SE_FWD_D"
I_RC[8]="NE_FWD_D"
I_RC[9]="CHT_FWD_D"
I_RC[10]="NW_FWD_I"
```

```
I_RC[11]="NC_FWD_I"
I_RC[12]="W_FWD_I"
I_RC[13]="S_FWD_I"
I_RC[14]="SE_FWD_I"
I_RC[15]="NE_FWD_I"
I_RC[16]="CHT_FWD_I"
I_RC[17]="NW_SW"
I_RC[18]="NC_SW"
I_RC[19]="W_SW"
I_RC[20]="S_SW"
I_RC[21]="SE_SW"
I_RC[22]="NE_SW"
I_RC[23]="CHT_SW"
I_RC[24]="NW_PT_D"
I_RC[25]="NC_PT_D"
I_RC[26]="W_PT_D"
I_RC[27]="S_PT_D"
I_RC[28]="SE_PT_D"
I_RC[29]="NE_PT_D"
I_RC[30]="CHT_PT_D"
I_RC[31]="NW_PT_I"
I_RC[32]="NC_PT_I"
I_RC[33]="W_PT_I"
I_RC[34]="S_PT_I"
I_RC[35]="SE_PT_I"
FOR X=1 TO 35
      APPEND BLANK
      REPL ITEM2 WITH I_RC[X]
NEXT
I_RC[1]="NE_PT_I"
I_RC[2]="CHT_PT_I"
I_RC[3]="NW_NP"
I_RC[4]="NC_NP"
I_RC[5]="W_NP"
I_RC[6]="S_NP"
I_RC[7]="SE_NP"
I_RC[8]="NE_NP"
I_RC[9]="CHT_NP"
I_RC[10]="NW_PPR"
I_RC[11]="NC_PPR"
I_RC[12]="W_PPR"
I_RC[13]="S_PPR"
I_RC[14]="SE_PPR"
I_RC[15]="NE_PPR"
I_RC[16]="CHT_PPR"
I_RC[17]="NW_PANEL"
I_RC[18]="NC_PANEL"
I_RC[19]="W_PANEL"
I_RC[20]="S_PANEL"
I_RC[21]="SE_PANEL"
I_RC[22]="NE_PANEL"
I_RC[23]="CHT_PANEL"
I_RC[24]="NF_T_NW"
I_RC[25]="NF_T_NC"
I_RC[26]="NF_T_W"
I_RC[27]="NF_T_S"
I_RC[28]="NF_T_SE"
I_RC[29]="NF_T_NE"
I_RC[30]="NF_T_CHT"
I_RC[31]="NF_P_NW"
I_RC[32]="NF_P_NC"
I_RC[33]="NF_P_W"
I_RC[34]="NF_P_S"
I_RC[35]="NF_P_SE"
FOR X=1 TO 35
      APPEND BLANK
      REPL ITEM2 WITH I_RC[X]
NEXT
I_RC[1]="NF_P_NE"
I_RC[2]="NF_P_CHT"
I_RC[3]="NF_F_NW"
I_RC[4]="NF_F_NC"
I_RC[5]="NF_F_W"
I_RC[6]="NF_F_S"
I_RC[7]="NF_F_SE"
I_RC[8]="NF_F_NE"
I_RC[9]="NF_F_CHT"
I_RC[10]="NF_PW_NW"
I_RC[11]="NF_PW_NC"
I_RC[12]="NF_PW_W"
I_RC[13]="NF_PW_S"
I_RC[14]="NF_PW_SE"
I_RC[15]="NF_PW_NE"
I_RC[16]="NF_PW_CHT"
I_RC[17]="PL_T_NW"
I_RC[18]="PL_T_NC"
I_RC[19]="PL_T_W"
I_RC[20]="PL_T_S"
I_RC[21]="PL_T_SE"
I_RC[22]="PL_T_NE"
I_RC[23]="PL_T_CHT"
I_RC[24]="PL_P_NW"
I_RC[25]="PL_P_NC"
I_RC[26]="PL_P_W"
I_RC[27]="PL_P_S"
I_RC[28]="PL_P_SE"
I_RC[29]="PL_P_NE"
I_RC[30]="PL_P_CHT"
I_RC[31]="PL_F_NW"
I_RC[32]="PL_F_NC"
I_RC[33]="PL_F_W"
I_RC[34]="PL_F_S"
I_RC[35]="PL_F_SE"
FOR X=1 TO 35
      APPEND BLANK
      REPL ITEM2 WITH I_RC[X]
NEXT
I_RC[1]="PL_F_NE"
I_RC[2]="PL_F_CHT"
I_RC[3]="PL_PW_NW"
I_RC[4]="PL_PW_NC"
I_RC[5]="PL_PW_W"
I_RC[6]="PL_PW_S"
I_RC[7]="PL_PW_SE"
I_RC[8]="PL_PW_NE"
I_RC[9]="PL_PW_CHT"
I_RC[10]="VF_T_NW"
I_RC[11]="VF_T_NC"
I_RC[12]="VF_T_W"
I_RC[13]="VF_T_S"
I_RC[14]="VF_T_SE"
I_RC[15]="VF_T_NE"
I_RC[16]="VF_T_CHT"
I_RC[17]="VF_P_NW"
I_RC[18]="VF_P_NC"
I_RC[19]="VF_P_W"
I_RC[20]="VF_P_S"
I_RC[21]="VF_P_SE"
I_RC[22]="VF_P_NE"
I_RC[23]="VF_P_CHT"
I_RC[24]="VF_F_NW"
I_RC[25]="VF_F_NC"
I_RC[26]="VF_F_W"
I_RC[27]="VF_F_S"
I_RC[28]="VF_F_SE"
I_RC[29]="VF_F_NE"
I_RC[30]="VF_F_CHT"
FOR X=1 TO 30
      APPEND BLANK
      REPL ITEM2 WITH I_RC[X]
NEXT
USE
```

```
RELEASE ALL LIKE I_*
RETURN .T.
*!**************************************
*!       Procedure: BY_INI
*!       Called by: SIM.PRG
*!       Calls: CR_IP2()           (func in SIM.PRG)
*!    Uses
*!         : &F_NAME1
*!         : &F_NAME2
*!    Indexes
*!         : &F_NAME1
*!         : &F_NAME2
*!    Memory Files
*!         : &F_NAME1
*!**************************************
PROC BY_INI
SET COLOR TO &CLR2
@22,0 CLEAR TO 24,79
@7,0 TO 9,17 DOUBLE
F_NAME="IP"+SPACE(5)
@8,3 SAY "FILE:" GET F_NAME
READ
IF F_NAME=" "
    F_NAME="IP"+SPACE(5)
    @8,3 SAY "FILE:" GET F_NAME
    CLEAR GETS
    F_NAME1="IP1"
    F_NAME2="IP2"
ELSE
    F_NAME1=ALLTRIM(F_NAME)+"1"
    F_NAME2=ALLTRIM(F_NAME)+"2"
ENDIF
IF .NOT.(FILE(F_NAME2+".DBF")
                        .OR.FILE(F_NAME1+".DBF"))
    F_ST="N"
    Y_B=1993
    Y_N=20
    Y_I=5
    Y_R=Y_N/Y_I+1
    Y_NF=6
    Y_PL=6
    SAVE ALL LIKE Y_* TO &F_NAME1
ELSE
    F_ST="O"
    IF !FILE(F_NAME1+".NTX")
         USE &F_NAME1
         INDEX ON ITEM1 TO &F_NAME1
    ENDIF
    IF !FILE(F_NAME2+".NTX")
         USE &F_NAME2
         INDEX ON ITEM2 TO &F_NAME2
    ENDIF
    USE
    RESTORE FROM &F_NAME1 ADDITIVE
ENDIF
@7,0 TO 9,79 DOUBLE
@7,17 SAY "┬"
@7,35 SAY "┬"
@7,59 SAY "┬"
@8,17 SAY "│"
@8,35 SAY "│"
@8,59 SAY "│"
@9,17 SAY "┴"
@9,35 SAY "┴"
@9,59 SAY "┴"
MY_B=Y_B
MY_N=Y_N
MY_I=Y_I
@ 8,19 SAY "BASE YEAR:" GET MY_B PICTURE
                               "9999" VALID MY_B#0
@ 8,37 SAY "YEARS TO SIMULATE:" GET MY_N
                          PICTURE "99" VALID MY_N#0
@ 8,61 SAY "YEAR INTERVAL:" GET MY_I
                          PICTURE "9" VALID MY_I#0
READ
IF LASTKEY()=27
    RETURN
ENDIF
IF MY_B#Y_B .OR. MY_N#Y_N .OR. MY_I#Y_I .OR.
F_ST="N"
    Y_B=MY_B
    Y_N=MY_N
    Y_I=MY_I
    Y_R=Y_N/Y_I+1
    Y_NF=6
    Y_PL=6
    SAVE ALL LIKE Y_* TO &F_NAME1
    CR_IP2(F_NAME1,F_NAME2)
ENDIF
DF_I1="Y"
S_SCR=SAVESCREEN(7,0,9,79)
FOR X=7 TO 21
    @X,0 CLEAR TO X+2,79
    RESTSCREEN(X+1,0,X+3,79,S_SCR)
    FOR Y=1 TO 50
    NEXT
NEXT
SAVE SCREEN TO SCR1
RETURN*!**************************************
*!     Procedure: D_INI
*!     Called by: DEMAND          (proc in SIM.PRG)
*!     Calls
*!          : INI_2   (procedure in SIM.PRG)
*!          : DB_2    (procedure in SIM.PRG)
*!          : INI_1   (procedure in SIM.PRG)
*!          : INI_3   (procedure in SIM.PRG)
*!          : DB_1    (procedure in SIM.PRG)
*!          : DB_3    (procedure in SIM.PRG)
*!     Uses
*!          : &F_NAME2
*!          : &F_NAME1
*!     Indexes
*!          : &F_NAME2
*!          : &F_NAME1
*!     Memory Files
*!          : &F_NAME1
*!**************************************
PROC D_INI
SELE A
DECLARE A_P_T[Y_R+1]
DECLARE A_PU1[Y_R+1]
DECLARE A_PU2[Y_R+1]
DECLARE A_PU3[Y_R+1]
DECLARE A_PU4[Y_R+1]
DECLARE A_PU5[Y_R+1]
DECLARE A_PU6[Y_R+1]
DECLARE A_PU7[Y_R+1]
DECLARE A_P11[Y_R+1]
DECLARE A_P12[Y_R+1]
DECLARE A_P13[Y_R+1]
DECLARE A_P14[Y_R+1]
DECLARE A_P21[Y_R+1]
DECLARE A_P22[Y_R+1]
DECLARE A_P23[Y_R+1]
DECLARE A_P24[Y_R+1]
DECLARE A_P31[Y_R+1]
DECLARE A_P32[Y_R+1]
DECLARE A_P33[Y_R+1]
DECLARE A_P34[Y_R+1]
DECLARE A_P41[Y_R+1]
DECLARE A_P42[Y_R+1]
DECLARE A_P43[Y_R+1]
DECLARE A_P44[Y_R+1]
DECLARE A_P51[Y_R+1]
DECLARE A_P52[Y_R+1]
DECLARE A_P53[Y_R+1]
DECLARE A_P54[Y_R+1]
DECLARE A_P61[Y_R+1]
DECLARE A_P62[Y_R+1]
DECLARE A_P63[Y_R+1]
DECLARE A_P64[Y_R+1]
DECLARE A_P71[Y_R+1]
DECLARE A_P72[Y_R+1]
DECLARE A_P73[Y_R+1]
DECLARE A_P74[Y_R+1]
DECLARE A_LIT[Y_R+1]
USE &F_NAME2 INDEX &F_NAME2
SEEK "T_POP"
DO INI_2 WITH A_P_T
SEEK "U_POP_1"
DO INI_2 WITH A_PU1
SEEK "U_POP_2"
DO INI_2 WITH A_PU2
SEEK "U_POP_3"
DO INI_2 WITH A_PU3
SEEK "U_POP_4"
DO INI_2 WITH A_PU4
SEEK "U_POP_5"
DO INI_2 WITH A_PU5
SEEK "U_POP_6"
DO INI_2 WITH A_PU6
SEEK "U_POP_7"
DO INI_2 WITH A_PU7
SEEK "NW_POP_1"
DO INI_2 WITH A_P11
SEEK "NW_POP_2"
DO INI_2 WITH A_P12
SEEK "NW_POP_3"
DO INI_2 WITH A_P13
SEEK "NW_POP_4"
DO INI_2 WITH A_P14
SEEK "NC_POP_1"
DO INI_2 WITH A_P21
SEEK "NC_POP_2"
DO INI_2 WITH A_P22
SEEK "NC_POP_3"
DO INI_2 WITH A_P23
SEEK "NC_POP_4"
DO INI_2 WITH A_P24
SEEK "W_POP_1"
DO INI_2 WITH A_P31
SEEK "W_POP_2"
DO INI_2 WITH A_P32
SEEK "W_POP_3"
DO INI_2 WITH A_P33
SEEK "W_POP_4"
DO INI_2 WITH A_P34
SEEK "S_POP_1"
DO INI_2 WITH A_P41
SEEK "S_POP_2"
DO INI_2 WITH A_P42
SEEK "S_POP_3"
DO INI_2 WITH A_P43
SEEK "S_POP_4"
DO INI_2 WITH A_P44
SEEK "SE_POP_1"
DO INI_2 WITH A_P51
SEEK "SE_POP_2"
DO INI_2 WITH A_P52
SEEK "SE_POP_3"
DO INI_2 WITH A_P53
SEEK "SE_POP_4"
DO INI_2 WITH A_P54
SEEK "NE_POP_1"
```

```
DO INI_2 WITH A_P61
SEEK "NE_POP_2"
DO INI_2 WITH A_P62
SEEK "NE_POP_3"
DO INI_2 WITH A_P63
SEEK "NE_POP_4"
DO INI_2 WITH A_P64
SEEK "CHT_POP_1"
DO INI_2 WITH A_P71
SEEK "CHT_POP_2"
DO INI_2 WITH A_P72
SEEK "CHT_POP_3"
DO INI_2 WITH A_P73
SEEK "CHT_POP_4"
DO INI_2 WITH A_P74
SEEK "LITERACY"
DO INI_2 WITH A_LIT
DO WHILE .T.
    @ 1,0 CLEAR TO 24, 79
    @ 1,0 TO 24, 79  DOUBLE
    @ 1,5 SAY "[ POPULATION ]"
    @ 3,3 SAY "YEAR  POPULATION
          LITERACY  ---- URBAN POPULATION
                         (% of regional total) ----"
    @ 4,3 SAY "      (in '000s)
                  (%)     N.WEST N.CEN.  WEST
                  SOUTH  S.EAST N.EAST  CHT"
    @ 5,3 SAY "----  ---------- --------
                               ------ ------ ------ -----
                                      ------ ------ -----"

    IF Y_DI="N"
        V=2
        FOR I=Y_B TO Y_B+Y_N STEP Y_I
            A_P_T[V]=(I*2010772.7261
                                  -3895455497.7)/1000
            A_LIT[V]=I*0.44400878
                                        -858.46764728

            V=V+1
        NEXT
    ENDIF
    @6,3 SAY STR(Y_B,4,0)
    @6,10 GET A_P_T[2] PICTURE
                                       "9999999"
    @6,20 GET A_LIT[2] PICTURE "999.999"
    FOR YY=1 TO 7
        Y=STR(YY,1,0)
        @6,23+YY*7 GET A_PU&Y[2]
                              PICTURE "99.999"
    NEXT
    IF F_ST="N"
        READ
        V=3
        FOR I=Y_B+Y_I TO Y_B+Y_N
                                         STEP Y_I

            FOR YY=1 TO 7
                Y=STR(YY,1,0)
                A_PU&Y[V]=(V-2)*Y_I*
                        0.1226+A_PU&Y[2]+0.1
            NEXT
            V=V+1
        NEXT
    ENDIF
    V=3
    X=7
    FOR I=Y_B+Y_I TO Y_B+Y_N
                                         STEP Y_I

        @X,3 SAY STR(I,4,0)
        @X,10 GET A_P_T[V] PICTURE  "9999999"

        @X,20 GET A_LIT[V] PICTURE  "999.999"
```

```
     FOR YY=1 TO 7
          Y=STR(YY,1,0)
          @X,23+YY*7 GET A_PU&Y[V]
                              PICTURE "99.999"
     NEXT
     X=X+1
     V=V+1
     IF X=23
          READ
          @6,2 CLEAR TO 23,78
          X=6
     ENDIF
NEXT
IF X>18
     READ
     @2,1 CLEAR TO 23,78
     X=2
ENDIF
@ X+1,3 SAY " ------------
            POPULATION DISTRIBUTION
              (as % of total population) ------"
@ X+2,3 SAY "  RICH
              N.WEST  N.CENT.  WEST
          SOUTH  S.EAST  N.EAST    CHT"
@ X+3,3 SAY " -------  -------
                        -------  -------  -------
                        -------  -------  -------"
@ X+4,5 GET A_P_T[1] PICTURE
                                  "999.999"
P='1'
FOR YY=1 TO 7
     Y=STR(YY,1,0)
     @X+4,5+YY*9 GET A_P&Y&P[1]
                          PICTURE "999.999"
NEXT
READ
PRMPT=1
@ 24,26 PROMPT " NEXT "
@ 24,35 PROMPT " EDIT "
@ 24,44 PROMPT " EXIT "
MENU TO PRMPT
DO CASE
CASE PRMPT=2
     LOOP
CASE PRMPT=3
     RELEASE ALL LIKE A_*
     RETURN
ENDCASE
@24, 26 SAY " UPDATING
                        DATA FILE...WAIT... "
Q='2'
R='3'
S='4'
FOR X=1 TO 7
     XX=STR(X,1,0)
     V=2
     FOR I=Y_B TO Y_B+Y_N STEP Y_I
          A_P&XX&P[V]=(A_P_T[V]*
               A_P&XX&P[1]/ 100)*(A_PU&XX[V]
                         / 100)*(A_P_T[1]/ 100)
          A_P&XX&Q[V]=(A_P_T[V]*
               A_P&XX&P[1]/ 100)*(A_PU&XX[V]
                    / 100)*(1-(A_P_T[1]/ 100))
          A_P&XX&R[V]=(A_P_T[V]*
           A_P&XX&P[1]/ 100)*(1-(A_PU&XX[V]
                        / 100))*(A_P_T[1]/ 100)
          A_P&XX&S[V]=(A_P_T[V]*
           A_P&XX&P[1]/ 100)*(1-(A_PU&XX[V]
                    / 100))*(1-(A_P_T[1]/ 100))
          V=V+1
     NEXT
```

```
NEXT
SEEK "T_POP"
DO DB_2 WITH A_P_T
SEEK "U_POP_1"
DO DB_2 WITH A_PU1
SEEK "U_POP_2"
DO DB_2 WITH A_PU2
SEEK "U_POP_3"
DO DB_2 WITH A_PU3
SEEK "U_POP_4"
DO DB_2 WITH A_PU4
SEEK "U_POP_5"
DO DB_2 WITH A_PU5
SEEK "U_POP_6"
DO DB_2 WITH A_PU6
SEEK "U_POP_7"
DO DB_2 WITH A_PU7
SEEK "NW_POP_1"
DO DB_2 WITH A_P11
SEEK "NW_POP_2"
DO DB_2 WITH A_P12
SEEK "NW_POP_3"
DO DB_2 WITH A_P13
SEEK "NW_POP_4"
DO DB_2 WITH A_P14
SEEK "NC_POP_1"
DO DB_2 WITH A_P21
SEEK "NC_POP_2"
DO DB_2 WITH A_P22
SEEK "NC_POP_3"
DO DB_2 WITH A_P23
SEEK "NC_POP_4"
DO DB_2 WITH A_P24
SEEK "W_POP_1"
DO DB_2 WITH A_P31
SEEK "W_POP_2"
DO DB_2 WITH A_P32
SEEK "W_POP_3"
DO DB_2 WITH A_P33
SEEK "W_POP_4"
DO DB_2 WITH A_P34
SEEK "S_POP_1"
DO DB_2 WITH A_P41
SEEK "S_POP_2"
DO DB_2 WITH A_P42
SEEK "S_POP_3"
DO DB_2 WITH A_P43
SEEK "S_POP_4"
DO DB_2 WITH A_P44
SEEK "SE_POP_1"
DO DB_2 WITH A_P51
SEEK "SE_POP_2"
DO DB_2 WITH A_P52
SEEK "SE_POP_3"
DO DB_2 WITH A_P53
SEEK "SE_POP_4"
DO DB_2 WITH A_P54
SEEK "NE_POP_1"
DO DB_2 WITH A_P61
SEEK "NE_POP_2"
DO DB_2 WITH A_P62
SEEK "NE_POP_3"
DO DB_2 WITH A_P63
SEEK "NE_POP_4"
DO DB_2 WITH A_P64
SEEK "CHT_POP_1"
DO DB_2 WITH A_P71
SEEK "CHT_POP_2"
DO DB_2 WITH A_P72
SEEK "CHT_POP_3"
DO DB_2 WITH A_P73
```

```
    SEEK "CHT_POP_4"
    DO DB_2 WITH A_P74
    SEEK "LITERACY"
    DO DB_2 WITH A_LIT
    RELEASE ALL LIKE A_*
    IF Y_DI="N"
        Y_DI="Y"
        SAVE ALL LIKE Y_*              TO &F_NAME1

    ENDIF
    EXIT
ENDDO
SELE B
USE &F_NAME1 INDEX &F_NAME1
DECLARE A_FWD1[5]
DECLARE A_FWD2[7]
DECLARE A_PT1[5]
DECLARE A_PT2[5]
DECLARE A_ST[7]
DECLARE A_NW[5]
DECLARE A_PA[5]
DECLARE A_PNL[7]
SEEK "FULWD_DU"
DO INI_1 WITH A_FWD1
SEEK "FULWD_IU"
DO INI_1 WITH A_FWD2
SEEK "PT/PL_DU"
DO INI_1 WITH A_PT1
SEEK "PT/PL_IU"
DO INI_1 WITH A_PT2
SEEK "STMBR"
DO INI_3 WITH A_ST
SEEK "NWPT"
DO INI_1 WITH A_NW
SEEK "PAPR"
DO INI_1 WITH A_PA
SEEK "PANEL"
DO INI_3 WITH A_PNL
DO WHILE .T.
    @ 0,0 CLEAR TO 24, 79
    @ 0,0 TO 24, 79   DOUBLE
    @ 0,5 SAY "[ CONSUMPTION IN
                THE YEAR "+STR(Y_B,4,0)+" ]"
    @ 2,2 SAY  "                ------
            PER CAPITA CONSUMPTION------|
            GOVT.& COMM.|CONSUMPTION"
    @ 3,2 SAY  "                |
                    --- URBAN----|----RURAL
        ----|CONSUMPTION |GROWTH RATE"
    @ 4,2 SAY  "          GROSS
            | RICH | POOR | RICH | POOR |
                    (% of urban)|(% per yr)"
    @ 5,2 SAY  "          ------|------|
                        ------|------|------|
                    -----------|-----------"
    @ 6,2 SAY  "FUELWOOD"
    @ 7,2 SAY  " -DOMESTIC  (m3):"
    FOR X=2 TO 5
        @7,13+X*7 GET A_FWD1[X]
    NEXT                    PICTURE "9.9999"
    @7,70 GET A_FWD1[1] PICTURE

    @ 8,2 SAY  " -INDUSTRIAL(m3):" GET    "999.999"
                A_FWD2[2] PICTURE "9.9999"
    @ 8,70 GET A_FWD2[1] PICTURE

    @ 9,2 SAY  "SAWN TIMBER (m3):"        "999.999"
    @10,2 SAY  " -RWE FACTOR (%):"
    FOR X=1 TO 2
        @9,20+X*7 GET A_ST[X] PICTURE
```

```
                                        "9.9999"
    NEXT
    @9,24+3*7 GET A_ST[3] PICTURE
                                        "9.9999"
    @9,58 GET A_ST[4] PICTURE "999.999"
    @9,70 GET A_ST[5] PICTURE "999.999"
    @10,20 GET A_ST[6] PICTURE "99.999"
    @11,2 SAY  "POSTS & POLES"
    @12,2 SAY  " -DOMESTIC  (m3):" GET
                    A_PT1[2] PICTURE "9.9999"
    @12,70 GET A_PT1[1] PICTURE
                                        "999.999"
    @13,2 SAY  " -INDUSTRIAL(m3):" GET
                    A_PT2[2] PICTURE "9.9999"
    @13,70 GET A_PT2[1] PICTURE
                                        "999.999"
    @17,2 SAY "                 NORTH
        NORTH                  SOUTH  NORTH"
    @18,2 SAY "                  WEST
                CENTRAL  WEST  SOUTH
                        EAST   EAST   CHT"
    @19,2 SAY "                 -------
                        ------- ------- ------
                        ------- ------- ------"

    @20,2 SAY  "PANEL PRODUCTS(m3):"
    FOR X=1 TO 7
        @20,14+X*8 GET A_PNL[X]
                            PICTURE "999.999"

    NEXT
    READ
    PRMPT=1
    @ 24,26 PROMPT " SAVE "
    @ 24,35 PROMPT " EDIT "
    @ 24,44 PROMPT " EXIT "
    MENU TO PRMPT
    DO CASE
    CASE PRMPT=2
        LOOP
    CASE PRMPT=3
        RELEASE ALL LIKE A_*
        RETURN
    ENDCASE
    @24, 26 SAY " UPDATING DATA
                            FILE...WAIT..."
    SEEK "FULWD_DU"
    DO DB_1 WITH A_FWD1
    SEEK "FULWD_IU"
    DO DB_1 WITH A_FWD2
    SEEK "PT/PL_DU"
    DO DB_1 WITH A_PT1
    SEEK "PT/PL_IU"
    DO DB_1 WITH A_PT2
    SEEK "STMBR"
    DO DB_3 WITH A_ST
    SEEK "NWPT"
    DO DB_1 WITH A_NW
    SEEK "PAPR"
    DO DB_1 WITH A_PA
    SEEK  "PANEL"
    DO DB_3 WITH A_PNL
    RELEASE ALL LIKE A_*
    EXIT
ENDDO
RETURN
*!*********************************************
*!      Procedure: INI_1
*!      Called by
*!          : D_INI          (procedure in SIM.PRG)
*!          : PAGE3          (procedure in SIM.PRG)
*!          : D_PR1          (procedure in SIM.PRG)
*!*********************************************
```

```
PROC INI_1
DECLARE A_VAR[5]
PARAMETERS A_VAR
A_VAR[1]=G_RATE1
A_VAR[2]=U_RICH
A_VAR[3]=U_POOR
A_VAR[4]=R_RICH
A_VAR[5]=R_POOR
RETURN
*!**************************************
*!   Procedure: DB_1
*!   Called by: D_INI          (procedure in SIM.PRG)
*!**************************************


PROC DB_1
DECLARE A_VAR[5]
PARAMETERS A_VAR
REPLACE G_RATE1 WITH A_VAR[1]
REPLACE U_RICH WITH A_VAR[2]
REPLACE U_POOR WITH A_VAR[3]
REPLACE R_RICH WITH A_VAR[4]
REPLACE R_POOR WITH A_VAR[5]
RETURN
*!**************************************
*!   Procedure: INI_2
*!   Called by
*!        : D_INI  (procedure in SIM.PRG)
*!        : D_VW1  (procedure in SIM.PRG)
*!        : D_PR1  (procedure in SIM.PRG)
*!**************************************
PROC INI_2
DECLARE A_VAR[Y_R+1]
PARAMETERS A_VAR
A_VAR[1]=G_RATE2
X=2
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    M_FNAME="Y"+STR(I,4,0)
    A_VAR[X]=&M_FNAME
    X=X+1
NEXT
RETURN
*!**************************************
*!   Procedure: DB_2
*!   Called by: D_INI          (procedure in SIM.PRG)
*!**************************************
PROC DB_2
DECLARE A_VAR[Y_R]
PARAMETERS A_VAR
REPLACE G_RATE2 WITH A_VAR[1]
X=2
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FLD="Y"+STR(I,4,0)
    REPLACE &FLD WITH A_VAR[X]
    X=X+1
NEXT
RETURN
*!**************************************
*!   Procedure: INI_3
*!   Called by
*!   : D_INI   (procedure in SIM.PRG)
*!   : PAGE3   (procedure in SIM.PRG)
*!   : VS_INI  (procedure in SIM.PRG)
*!   : S_SIM   (procedure in SIM.PRG)
*!   : S_VW1   (procedure in SIM.PRG)
*!   : D_PR1   (procedure in SIM.PRG)
*!   : S_PR1   (procedure in SIM.PRG)
*!**************************************PROCINI_3
DECLARE A_VAR[7]
PARAMETERS A_VAR
A_VAR[1]=N_WEST
```

```
A_VAR[2]=N_CENT
A_VAR[3]=WEST
A_VAR[4]=SOUTH
A_VAR[5]=S_EAST
A_VAR[6]=N_EAST
A_VAR[7]=CHT
RETURN
*!**************************************
*!   Procedure: DB_3
*!   Called by
*!   : D_INI       (procedure in SIM.PRG)
*!   : VS_INI  (procedure in SIM.PRG)
*!**************************************
PROC DB_3
DECLARE A_VAR[7]
PARAMETERS A_VAR
REPLACE N_WEST WITH A_VAR[1]
REPLACE N_CENT WITH A_VAR[2]
REPLACE WEST WITH A_VAR[3]
REPLACE SOUTH WITH A_VAR[4]
REPLACE S_EAST WITH A_VAR[5]
REPLACE N_EAST WITH A_VAR[6]
REPLACE CHT WITH A_VAR[7]
RETURN
*!**************************************
*!   Procedure: D_SIM
*!   Called by: DEMAND          (proc in SIM.PRG)
*!   Uses
*!        : &F_NAME2
*!        : &F_NAME1
*!   Indexes
*!        : &F_NAME2
*!        : &F_NAME1
*!**************************************
PROC D_SIM
@13,10 TO 17,69 DOUBLE
@15,22 SAY "SIMULATION IN
                              PROGRESS....WAIT....."
SELE A
USE &F_NAME2 INDEX &F_NAME2
SELE B
USE &F_NAME1 INDEX &F_NAME1
DECLARE A_P[4]
DECLARE A_C[4]
DECLARE A_G[4]
DECLARE A_S[7]
DECLARE A_S1[7]
A_S[1]="NW_POP"
A_S[2]="NC_POP"
A_S[3]="W_POP"
A_S[4]="S_POP"
A_S[5]="SE_POP"
A_S[6]="NE_POP"
A_S[7]="CHT_POP"
A_S1[1]="NW_FWD_D"
A_S1[2]="NC_FWD_D"
A_S1[3]="W_FWD_D"
A_S1[4]="S_FWD_D"
A_S1[5]="SE_FWD_D"
A_S1[6]="NE_FWD_D"
A_S1[7]="CHT_FWD_D"
SELE B
SEEK "FULWD_DU"
A_G[1]=G_RATE1
A_C[1]=U_RICH
A_C[2]=U_POOR
A_C[3]=R_RICH
A_C[4]=R_POOR
FOR V=1 TO 7
    Y=1
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
```

```
        FLD="Y"+STR(I,4,0)
        SELE A
        SEEK A_S[V]
        FOR X=1 TO 4
             A_P[X]=&FLD
             SKIP
        NEXT
        A_V=A_P[1]*A_C[1]*(1+A_G[1]
                     / 100)^(Y*Y_I-Y_I)+A_P[2]*A_C[2]
                                     *(1+A_G[1]/ 100)^(Y*Y_I-Y_I)+A_P[3]
             *(1+A_G[1]/ 100)^(Y*Y_I-Y_I)
              *A_C[3]*(1+A_G[1]/ 100)^(Y*Y_I-Y_I)
                              +A_P[4]*A_C[4]*(1+A_G[1]
                                      / 100)^(Y*Y_I-Y_I)

        SELE A
        SEEK A_S1[V]
        REPLACE &FLD WITH A_V
        Y=Y+1
    NEXT
NEXT
A_S1[1]="NW_FWD_I"
A_S1[2]="NC_FWD_I"
A_S1[3]="W_FWD_I"
A_S1[4]="S_FWD_I"
A_S1[5]="SE_FWD_I"
A_S1[6]="NE_FWD_I"
A_S1[7]="CHT_FWD_I"
SELE B
SEEK "FULWD_IU"
A_G[1]=G_RATE1
A_C[1]=U_RICH
FOR V=1 TO 7
    Y=1
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
        FLD="Y"+STR(I,4,0)
        SELE A
        SEEK A_S[V]
        FOR X=1 TO 4
             A_P[X]=&FLD
             SKIP
        NEXT
        A_V=(A_P[1]+A_P[2]+A_P[3]
                     +A_P[4])*A_C[1]*(1+A_G[1]
                                     / 100)^(Y*Y_I-Y_I)

        SELE A
        SEEK A_S1[V]
        REPLACE &FLD WITH A_V
        Y=Y+1
    NEXT
NEXT
A_S1[1]="NW_SW"
A_S1[2]="NC_SW"
A_S1[3]="W_SW"
A_S1[4]="S_SW"
A_S1[5]="SE_SW"
A_S1[6]="NE_SW"
A_S1[7]="CHT_SW"
SELE B
SEEK "STMBR"
A_C[1]=N_WEST
A_C[2]=N_CENT
A_C[3]=WEST
A_C[4]=SOUTH
A_G[1]=S_EAST
A_G[2]=N_EAST
A_G[3]=CHT
FOR V=1 TO 7
    Y=1
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
        FLD="Y"+STR(I,4,0)
        SELE A
        SEEK A_S[V]
```

```
            FOR X=1 TO 4
                 A_P[X]=&FLD
                 SKIP
            NEXT
            A_V=A_P[1]*A_C[1]*
                         ((1+A_G[1]/ 100)^(Y*Y_I-Y_I)+
                              A_C[4]/ 100)+A_P[2]*A_C[2]*
                         ((1+A_G[1]/ 100)^(Y*Y_I-Y_I)+
                          A_C[4]/ 100)+(A_P[3]+A_P[4])*
                  A_C[3]*(1+A_G[1]/ 100)^(Y*Y_I-Y_I)

            SELE A
            SEEK A_S1[V]
            IF A_G[2]# 0
                 REPLACE &FLD WITH
                                          A_V*100/ A_G[2]

            ENDIF
            Y=Y+1
        NEXT
NEXT
A_S1[1]="NW_PT_D"
A_S1[2]="NC_PT_D"
A_S1[3]="W_PT_D"
A_S1[4]="S_PT_D"
A_S1[5]="SE_PT_D"
A_S1[6]="NE_PT_D"
A_S1[7]="CHT_PT_D"
SELE B
SEEK "PT/ PL_DU"
A_G[1]=G_RATE1
A_C[1]=U_RICH
FOR V=1 TO 7
     Y=1
     FOR I=Y_B TO Y_B+Y_N STEP Y_I
          FLD="Y"+STR(I,4,0)
          SELE A
          SEEK A_S[V]
          SKIP
          SKIP
          FOR X=1 TO 2
               A_P[X]=&FLD
               SKIP
          NEXT
          A_V=(A_P[1]+A_P[2])*
                              A_C[1]*(1+A_G[1]/ 100)
                                          ^(Y_I*(Y-1))

          SELE A
          SEEK A_S1[V]
          REPLACE &FLD WITH A_V
          Y=Y+1
     NEXT
NEXT
A_S1[1]="NW_PT_I"
A_S1[2]="NC_PT_I"
A_S1[3]="W_PT_I"
A_S1[4]="S_PT_I"
A_S1[5]="SE_PT_I"
A_S1[6]="NE_PT_I"
A_S1[7]="CHT_PT_I"
SELE B
SEEK "PT/ PL_IU"
A_G[1]=G_RATE1
A_C[1]=U_RICH
FOR V=1 TO 7
     Y=1
     FOR I=Y_B TO Y_B+Y_N STEP Y_I
          FLD="Y"+STR(I,4,0)
          SELE A
          SEEK A_S[V]
          FOR X=1 TO 4
               A_P[X]=&FLD
               SKIP
            NEXT
            A_V=(A_P[1]+A_P[2]+
                              A_P[3]+A_P[4])*A_C[1]*
                        (1+A_G[1]/100)^(Y_I*(Y-1))
            SELE A
            SEEK A_S1[V]
            REPLACE &FLD WITH A_V
            Y=Y+1
        NEXT
    NEXT
    A_S1[1]="NW_NP"
    A_S1[2]="NC_NP"
    A_S1[3]="W_NP"
    A_S1[4]="S_NP"
    A_S1[5]="SE_NP"
    A_S1[6]="NE_NP"
    A_S1[7]="CHT_NP"
    A_G[1]=.0015772016
    A_G[2]=.0044214406
    A_C[1]=-123216.7921
    A_C[2]=-73405.40809
    SELE A
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
        FLD="Y"+STR(I,4,0)
        SEEK "T_POP"
        A_P[1]=&FLD
        SEEK "LITERACY"
        A_P[2]=&FLD
        A_C[3]=A_P[1]*1000*A_G[1]+A_C[1]
        A_C[4]=A_P[1]*A_P[2]/100*
                                   1000*A_G[2]+A_C[2]
        SEEK A_S1[1]
        REPLACE &FLD WITH A_C[3]/1000
        SEEK A_S1[2]
        REPLACE &FLD WITH A_C[4]/1000
    NEXT
    A_S1[1]="NW_PPR"
    A_S1[2]="NC_PPR"
    A_S1[3]="W_PPR"
    A_S1[4]="S_PPR"
    A_S1[5]="SE_PPR"
    A_S1[6]="NE_PPR"
    A_S1[7]="CHT_PPR"
    A_G[1]=.0007876442
    A_G[2]=.0022079712
    A_C[1]=-17378.69375
    A_C[2]=7498.3082248
    SELE A
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
        FLD="Y"+STR(I,4,0)
        SEEK "T_POP"
        A_P[1]=&FLD
        SEEK "LITERACY"
        A_P[2]=&FLD
        A_C[3]=A_P[1]*1000*A_G[1]+A_C[1]
        A_C[4]=A_P[1]*A_P[2]/100*
                                   1000*A_G[2]+A_C[2]
        SEEK A_S1[1]
        REPLACE &FLD WITH A_C[3]/1000
        SEEK A_S1[2]
        REPLACE &FLD WITH A_C[4]/1000
    NEXT
    A_S1[1]="NW_PANEL"
    A_S1[2]="NC_PANEL"
    A_S1[3]="W_PANEL"
    A_S1[4]="S_PANEL"
    A_S1[5]="SE_PANEL"
    A_S1[6]="NE_PANEL"
    A_S1[7]="CHT_PANEL"
    SELE B
    SEEK "PANEL"
```

```
A_G[1]=G_RATE1
FOR V=1 TO 7
    Y=1
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
        FLD="Y"+STR(I,4,0)
        SELE B
        FLD1=FIELD(V+2)
        A_P[1]=&FLD1
        A_V=A_P[1]*(1+A_G[1]
                              /100)^(Y*Y_I-Y_I)
        SELE A
        SEEK A_S1[V]
        REPLACE &FLD WITH A_V
        Y=Y+1
    NEXT
NEXT
RELEASE ALL LIKE A_*
CLOSE DATABASES
RETURN
*!***************************************
*!    Procedure: D_VW
*!    Called by: DEMAND          (proc in SIM.PRG)
*!    Calls
*!          : Z_NOTE    (procedure in SIM.PRG)
*!          : D_VW1 (procedure in SIM.PRG)
*!          : D_VW2 (procedure in SIM.PRG)
*!***************************************
PROC D_VW
DO WHILE D2#0
    @9,31 TO 12,52 DOUBLE
    @10,32 PROMPT " INITIAL
                              PARAMETERS "
    @11,32 PROMPT " SIMULATION
                                  RESULTS "
    MENU TO D2
    IF D2#0
        SAVE SCREEN TO SCR3
        DO CASE
        CASE D2=1
            DO Z_NOTE
            DO D_VW1
        CASE D2=2
            DO D_VW2
        ENDCASE
        CLOSE DATABASES
        REST SCREEN FROM SCR3
    ENDIF
ENDDO
RETURN
*!***************************************
*!    Procedure: D_VW1
*!    Called by: D_VW         (procedure in SIM.PRG)
*!    Calls
*!          : INI_2   (procedure in SIM.PRG)
*!          : &PROX
*!    Uses
*!          : &F_NAME2
*!          : &F_NAME1
*!    Indexes
*!          : &F_NAME2
*!          : &F_NAME1
*!***************************************
PROC D_VW1
SELE A
USE &F_NAME2 INDEX &F_NAME2
SELE B
USE &F_NAME1 INDEX &F_NAME1
DECLARE A_P_T[Y_R+1]
DECLARE A_PU1[Y_R+1]
DECLARE A_PU2[Y_R+1]
DECLARE A_PU3[Y_R+1]
DECLARE A_PU4[Y_R+1]
DECLARE A_PU5[Y_R+1]
DECLARE A_PU6[Y_R+1]
DECLARE A_PU7[Y_R+1]
DECLARE A_LIT[Y_R+1]
DECLARE A_P11[Y_R+1]
DECLARE A_P12[Y_R+1]
DECLARE A_P13[Y_R+1]
DECLARE A_P14[Y_R+1]
DECLARE A_P21[Y_R+1]
DECLARE A_P22[Y_R+1]
DECLARE A_P23[Y_R+1]
DECLARE A_P24[Y_R+1]
DECLARE A_P31[Y_R+1]
DECLARE A_P32[Y_R+1]
DECLARE A_P33[Y_R+1]
DECLARE A_P34[Y_R+1]
DECLARE A_P41[Y_R+1]
DECLARE A_P42[Y_R+1]
DECLARE A_P43[Y_R+1]
DECLARE A_P44[Y_R+1]
DECLARE A_P51[Y_R+1]
DECLARE A_P52[Y_R+1]
DECLARE A_P53[Y_R+1]
DECLARE A_P54[Y_R+1]
DECLARE A_P61[Y_R+1]
DECLARE A_P62[Y_R+1]
DECLARE A_P63[Y_R+1]
DECLARE A_P64[Y_R+1]
DECLARE A_P71[Y_R+1]
DECLARE A_P72[Y_R+1]
DECLARE A_P73[Y_R+1]
DECLARE A_P74[Y_R+1]
DECLARE A_LIT[Y_R+1]
SELE A
SEEK "T_POP"
DO INI_2 WITH A_P_T
SEEK "U_POP_1"
DO INI_2 WITH A_PU1
SEEK "U_POP_2"
DO INI_2 WITH A_PU2
SEEK "U_POP_3"
DO INI_2 WITH A_PU3
SEEK "U_POP_4"
DO INI_2 WITH A_PU4
SEEK "U_POP_5"
DO INI_2 WITH A_PU5
SEEK "U_POP_6"
DO INI_2 WITH A_PU6
SEEK "U_POP_7"
DO INI_2 WITH A_PU7
SEEK "LITERACY"
DO INI_2 WITH A_LIT
SEEK "NW_POP_1"
DO INI_2 WITH A_P11
SEEK "NW_POP_2"
DO INI_2 WITH A_P12
SEEK "NW_POP_3"
DO INI_2 WITH A_P13
SEEK "NW_POP_4"
DO INI_2 WITH A_P14
SEEK "NC_POP_1"
DO INI_2 WITH A_P21
SEEK "NC_POP_2"
DO INI_2 WITH A_P22
SEEK "NC_POP_3"
DO INI_2 WITH A_P23
SEEK "NC_POP_4"
DO INI_2 WITH A_P24
SEEK "W_POP_1"
DO INI_2 WITH A_P31
SEEK "W_POP_2"
DO INI_2 WITH A_P32
SEEK "W_POP_3"
DO INI_2 WITH A_P33
SEEK "W_POP_4"
DO INI_2 WITH A_P34
SEEK "S_POP_1"
DO INI_2 WITH A_P41
SEEK "S_POP_2"
DO INI_2 WITH A_P42
SEEK "S_POP_3"
DO INI_2 WITH A_P43
SEEK "S_POP_4"
DO INI_2 WITH A_P44
SEEK "SE_POP_1"
DO INI_2 WITH A_P51
SEEK "SE_POP_2"
DO INI_2 WITH A_P52
SEEK "SE_POP_3"
DO INI_2 WITH A_P53
SEEK "SE_POP_4"
DO INI_2 WITH A_P54
SEEK "NE_POP_1"
DO INI_2 WITH A_P61
SEEK "NE_POP_2"
DO INI_2 WITH A_P62
SEEK "NE_POP_3"
DO INI_2 WITH A_P63
SEEK "NE_POP_4"
DO INI_2 WITH A_P64
SEEK "CHT_POP_1"
DO INI_2 WITH A_P71
SEEK "CHT_POP_2"
DO INI_2 WITH A_P72
SEEK "CHT_POP_3"
DO INI_2 WITH A_P73
SEEK "CHT_POP_4"
DO INI_2 WITH A_P74
@ 0,0 CLEAR TO 24,79
@ 0,0 TO 24,79 DOUBLE
@ 0,72 SAY "PgUp-"+CHR(24)
@ 24,72 SAY "PgDn-"+CHR(25)
@ 24,2 SAY "Esc-exit"
@0,20 SAY ' INITIAL PARAMETERS FOR
                                DEMAND SIMULATION'
LX=1
P_CH=1
ST=0
PG=1
I_P=0
I_P1=Y_B
V1=2
G=LTRIM(STR(PG,2,0))
V=0


DO WHILE .T.
      @ 0,5 TO 0,15 DOUBLE
      @ 0,5 SAY "Page "+G
      PROX='PAGE'+LTRIM
                                (STR(P_CH,2,0))
      DO &PROX
      KEY=0
      DO WHILE .T.
            KEY=INKEY()
            DO CASE
            CASE KEY=27
                  RETURN
            CASE KEY=3.OR.KEY=18
                  EXIT
            ENDCASE
```

```
     ENDDO
     DO CASE
     CASE KEY=18
          PG=PG-1
     CASE KEY=3
          PG=PG+1
     ENDCASE
     G=LTRIM(STR(PG,2,0))
     IF I_P=Y_B+Y_N+Y_I
          IF KEY=18
               GX=LTRIM(STR(PG+1,2,0))
               IF I_P&GX=Y_B
                    P_CH=P_CH-1
               ENDIF
          ELSE
               I_P&G=Y_B
               V&G=2
               P_CH=P_CH+1
          ENDIF
     ELSE
          IF KEY=18
               GX=LTRIM(STR(PG+1,2,0))
               IF I_P&GX=Y_B
                    P_CH=P_CH-1
               ENDIF
          ELSE
               I_P&G=I_P
               V&G=V
          ENDIF
     ENDIF
     IF PG=0.OR.P_CH=0.OR.P_CH=4
          RETURN
     ENDIF
     @1,1 CLEAR TO 23,78
ENDDO
RELEASE ALL LIKE A_*
RETURN*!*************************************
*!   Procedure: PAGE1
*!*************************************
PROC PAGE1
@ 1,5 SAY "[ POPULATION ]"
@ 3,3 SAY "YEAR  POPULATION
            LITERACY  ---- URBAN POPULATION
                           (% of regional total) ----"
@ 4,3 SAY "     (in '000s)  (%)
               N.WEST N.CEN.  WEST  SOUTH
                           S.EAST N.EAST  CHT"
@ 5,3 SAY "----  --------- -------

                              ------ ------ ------ ------
                                     ------ ------ ------"
X=6
V=V&G
FOR I=I_P&G TO Y_B+Y_N STEP Y_I
     @X,3 SAY STR(I,4,0)
     @X,10 SAY A_P_T[V] PICTURE
                                             "9999999"
     @X,20 SAY A_LIT[V] PICTURE
                                             "999.999"
     FOR YY=1 TO 7
          Y=STR(YY,1,0)
          @X,23+YY*7 SAY A_PU&Y[V]
                                    PICTURE "99.999"
     NEXT
     X=X+1
     V=V+1
     IF X=20
          I_P=I+Y_I
          RETURN
     ENDIF
NEXT
@ X+1,3 SAY " ----------- POPULATION
```

```
                         DISTRIBUTION (as % of total
                                      population) ------"
@ X+2,3 SAY "  RICH   N.WEST
                     N.CENT.  WEST   SOUTH
                         S.EAST  N.EAST   CHT"
@ X+3,3 SAY " -------  -------

                                -------  -------  -------
                                 -------  -------  -------"
@ X+4,5 SAY A_P_T[1] PICTURE "999.999"
P='1'
FOR YY=1 TO 7
     Y=STR(YY,1,0)
     @X+4,5+YY*9 SAY A_P&Y&P[1]
                                PICTURE "999.999"
NEXT
I_P=Y_B+Y_N+Y_I
RETURN


*!*************************************
*!    Procedure: PAGE2
*!*************************************
PROC PAGE2
@ 1,5 SAY "[ POPULATION ('000s) ]"
@ 3,3 SAY "              NORTH
           NORTH              SOUTH  NORTH"
@ 4,3 SAY "YEAR             WEST
                    CENTRAL  WEST  SOUTH
                             EAST   EAST   CHT"
@ 5,3 SAY "----           ------- -------
                    ------- ------- ------- ------- -------"
XX=5
V=V&G
FOR I=I_P&G TO Y_B+Y_N STEP Y_I
     @XX+1,3 SAY STR(I,4,0) +
                               "  URBAN-RICH:"
     @XX+2,3 SAY "            -POOR:"
     @XX+3,3 SAY "       RURAL-RICH:"
     @XX+4,3 SAY "            -POOR:"
     @XX+5,3 SAY "  ------- ------- ------- ------- -------"
     FOR X=1 TO 7
          P=STR(X,1,0)
          FOR Y=1 TO 4
               YY=STR(Y,1,0)
               @XX+Y,14+X*8 SAY
                    A_P&P&YY[V] PICTURE "9999999"
          NEXT
     NEXT
     V=V+1
     XX=XX+5
     IF XX=20
          I_P=I+Y_I
          RETURN
     ENDIF
NEXT
I_P=Y_B+Y_N+Y_I
RETURN
*!*************************************
*!    Procedure: PAGE3
*!    Calls
*!    : INI_1   (procedure in SIM.PRG)
*!    : INI_3   (procedure in SIM.PRG)
*!*************************************
PROC PAGE3
SELE B
DECLARE A_FWD1[5]
DECLARE A_FWD2[5]
DECLARE A_PT1[5]
DECLARE A_PT2[5]
DECLARE A_ST[7]
DECLARE A_NW[5]
DECLARE A_PA[5]
DECLARE A_PNL[7]
SEEK "FULWD_DU"
DO INI_1 WITH A_FWD1
SEEK "FULWD_IU"
DO INI_1 WITH A_FWD2
SEEK "PT/PL_DU"
DO INI_1 WITH A_PT1
SEEK "PT/PL_IU"
DO INI_1 WITH A_PT2
SEEK "STMBR"
DO INI_3 WITH A_ST
SEEK "NWPT"
DO INI_1 WITH A_NW
SEEK "PAPR"
DO INI_1 WITH A_PA
SEEK "PANEL"
DO INI_3 WITH A_PNL
@  0,0  CLEAR TO 24, 79
@ 0,0  TO 24, 79   DOUBLE
@  1,5  SAY  "[  CONSUMPTION  IN  THE  YEAR
"+STR(Y_B,4,0) + " ]"
@ 3,2 SAY  "                    ------PER
                  CAPITA CONSUMPTION------|
            GOVT.& COMM.|CONSUMPTION"
                            |
@ 4,2 SAY  "
                      --- URBAN----|----RURAL----|
            CONSUMPTION |GROWTH RATE"
                        GROSS |
@ 5,2 SAY  "
                  RICH | POOR | RICH | POOR |
                        (% of urban)|(% per yr)"
@ 6,2 SAY  "              ------|------|------|
                  ------|------|-----------|----------"

@ 7,2 SAY  "FUELWOOD"
@ 8,2 SAY  " -DOMESTIC  (m3):"
FOR X=2 TO 5
    @8,13+X*7 SAY A_FWD1[X] PICTURE
                                        "9.9999"
NEXT
@8,70 SAY A_FWD1[1] PICTURE "999.999"
@ 9,2 SAY  " -INDUSTRIAL(m3):
                         "+STR(A_FWD2[2],6,4)
@9,70 SAY A_FWD2[1] PICTURE "999.999"
@10,2 SAY  "SAWN TIMBER (m3):"
@11,2 SAY  " -RWE FACTOR (%):"
FOR X=1 TO 2
    @10,20+X*7 SAY A_ST[X] PICTURE
                                        "9.9999"
NEXT
@10,24+3*7 SAY A_ST[3] PICTURE "9.9999"
@10,58 SAY A_ST[4] PICTURE "999.999"
@10,70 SAY A_ST[5] PICTURE "999.999"
@11,20 SAY A_ST[6] PICTURE "99.999"
@12,2 SAY "POSTS & POLES"

@13,2 SAY " -DOMESTIC  (m3): "
                              +STR(A_PT1[2],6,4)
@13,70 SAY A_PT1[1] PICTURE "999.999"
@14,2 SAY " -INDUSTRIAL(m3): "
                              +STR(A_PT2[2],6,4)
@14,70 SAY A_PT2[1] PICTURE "999.999"
@15,2 SAY "NEWSPRINT  (kg): "
                              +STR(A_NW[2],6,4)
@15,70 SAY A_NW[1] PICTURE "999.999"
@16,2 SAY "OTHER PAPERS(kg): "
                              +STR(A_PA[2],6,4)
@16,70 SAY A_PA[1] PICTURE "999.999"
@18,2 SAY "                NORTH
          NORTH                   SOUTH   NORTH"
@19,2 SAY "                WEST
                CENTRAL   WEST   SOUTH
                                   EAST    EAST    CHT"
                               ------- -------
@20,2 SAY "               ------- ------- ------- ------- -------"
@21,2 SAY "PANEL PRODUCTS(m3):"
FOR X=1 TO 7
    @21,14+X*8 SAY A_PNL[X] PICTURE
                                        "999.999"

NEXT
RETURN
*!*************************************************
*!    Procedure: D_VW2            (procedure in SIM.PRG)
*!    Called by: D_VW
*!    Calls: &PROX
*!    Uses: &F_NAME2
*!    Indexes: &F_NAME2
*!*************************************************
PROC D_VW2
SELE A
USE &F_NAME2 INDEX &F_NAME2
@ 0,0 CLEAR TO 24,79
@ 0,0 TO 24,79 DOUBLE
@ 0,72 SAY "PgUp-"+CHR(24)
@ 24,72 SAY "PgDn-"+CHR(25)
@ 24,2 SAY "Esc-exit"
@0,20  SAY  '  FUTURE  PROJECTIONS  FROM
DEMAND SIMULATION '
HD1="            N.WEST N.CENTR.  WEST
                                  SOUTH   S.EAST
                    N.EAST   CHT   NATIONAL"

HD2="          ------- ------- ------- -------
                              ------- ------- ------- -------"

LX=1
ST=0
PG=1
P_CH=.T.
PGS=1
PGN=1
M_I1=Y_B
M_I=Y_B
G='1'
DECLARE A_S1[7]
DECLARE TOT[Y_N+1]
FOR Z=1 TO Y_N+1
    TOT[Z]=0
NEXT
I=0
X1=2
X=5
PGS1=1
T='1'
CH_PGN=.T.
PGS_E=9
DO WHILE .T.
    @0,5 TO 0,15 DOUBLE
    @0,5 SAY "Page "+T
    PROX='PGS'+LTRIM(STR(PGS,2,0))
    DO &PROX
    IF X>18.OR.PGS=PGS_E-1
        KEY=0
        DO WHILE .T.
            KEY=INKEY()
            DO CASE
            CASE KEY=27
                RETURN
            CASE KEY=3.OR.KEY=18
                EXIT
            ENDCASE
        ENDDO
    ELSE
        KEY=3
    ENDIF
    DO CASE
    CASE KEY=18
         IF !CH_PGN
              DO  WHILE X&G#2
                   PG=PG-1
                   G=LTRIM(STR(PG,2,0))
              ENDDO
         ENDIF
         PG=PG-1
         G=LTRIM(STR(PG,2,0))
         IF PG=0
              RETURN
         ENDIF
         DO WHILE X&G#2
              PG=PG-1
              G=LTRIM(STR(PG,2,0))
         ENDDO
         PGN=PGN-1
         T=LTRIM(STR(PGN,2,0))
         PGS=PGS&T
    CASE KEY=3
         PG=PG+1
         G=LTRIM(STR(PG,2,0))
         IF P_CH
              FOR Z=1 TO Y_N+1
                   TOT[Z]=0
              NEXT
              PGS=PGS+1
              IF PGS_E=16.AND.PGS=8
                   PGS=9
              ENDIF
              M_I&G=Y_B
              M_I=Y_B
         ELSE
              X=X+1
              M_I&G=M_I
         ENDIF
         IF X>18
              X&G=2
              X=5
              PGN=PGN+1
              T=LTRIM(STR(PGN,2,0))
              PGS&T=PGS
              CH_PGN=.T.
         ELSE
              X=X+1
              X&G=X
              CH_PGN=.F.
         ENDIF
    ENDCASE
    IF PG=0.OR.PGS=PGS_E
         RETURN
    ENDIF
    @X&G,1 CLEAR TO 23,78
ENDDO
RETURN
*!*************************************
*!    Procedure: I_TOT
*!    Called by
*!         : PGS7    (procedure in SIM.PRG)
*!         : PGS8    (procedure in SIM.PRG)
*!         : D_PR2X(procedure in SIM.PRG)
*!         : D_PR2Y(procedure in SIM.PRG)
*!         : S_PR2X (procedure in SIM.PRG)
*!*************************************
PROC I_TOT
FOR Z=1 TO Y_N+1
    TOT[Z]=0
NEXT
RETURN
```

```
*!*************************************
*!    Procedure: PGS1
*!    Calls: D_V_R1      (procedure in SIM.PRG)
*!*************************************
PROC PGS1
A_S1[1]="NW_FWD_D"
A_S1[2]="NC_FWD_D"
A_S1[3]="W_FWD_D"
A_S1[4]="S_FWD_D"
A_S1[5]="SE_FWD_D"
A_S1[6]="NE_FWD_D"
A_S1[7]="CHT_FWD_D"
@X&G,3 SAY  "FUELWOOD -
                                    DOMESTIC USE"
@X&G+1,3 SAY  "---------------------"
DO D_V_R1
RETURN
*!*************************************
*!     Procedure: D_V_R1
*!          Called by
*!          : PGS1    (procedure in SIM.PRG)
*!          : PGS2    (procedure in SIM.PRG)
*!          : PGS3    (procedure in SIM.PRG)
*!          : PGS4    (procedure in SIM.PRG)
*!          : PGS5    (procedure in SIM.PRG)
*!          : PGS6    (procedure in SIM.PRG)
*!*************************************
PROC D_V_R1
@X&G,66 SAY "(in '000 m3)"
@X&G+2,3 SAY HD1
@X&G+3,3 SAY HD2
FOR V=1 TO 8
     X=X&G+4
     IF V#8
          SEEK A_S1[V]
     ENDIF
     FOR I=M_I&G TO IIF(M_I&G+
               (18-X&G)*Y_I>=Y_B+Y_N,Y_B+Y_N,
                         M_I&G+(18-X&G)*Y_I)
                                          STEP Y_I
          IF V#8
               FLD="Y"+STR(I,4,0)
               IF V=1
                    @X,3 SAY "YEAR-"+
                                          STR(I,4,0)
               ENDIF
               @X,6+8*V SAY &FLD
                                   PICTURE "9999.99"
               TOT[(I-Y_B)/Y_I+1]=
                         TOT[(I-Y_B)/Y_I+1]+&FLD
          ELSE
               @X,6+8*V SAY TOT[(I-Y_B)
                         /Y_I+1] PICTURE "99999.99"
          ENDIF
          X=X+1
     NEXT
NEXT
IF I<Y_B+Y_N+Y_I
     M_I=I
     P_CH=.F.
ELSE
     P_CH=.T.
ENDIF
RETURN
*!*************************************
*!    Procedure: PGS2
*!    Calls: D_V_R1      (procedure in SIM.PRG)
*!*************************************
PROC PGS2
A_S1[1]="NW_FWD_I"
A_S1[2]="NC_FWD_I"
```

```
A_S1[3]="W_FWD_I"
A_S1[4]="S_FWD_I"
A_S1[5]="SE_FWD_I"
A_S1[6]="NE_FWD_I"
A_S1[7]="CHT_FWD_I"
@X&G,3 SAY "FUELWOOD
                                - INDUSTRIAL USE"
@X&G+1,3 SAY "-----------------------"
DO D_V_R1
RETURN
*!************************************
*!   Procedure: PGS3
*!   Calls: D_V_R1(procedure in SIM.PRG)
*!************************************
PROC PGS3
A_S1[1]="NW_SW"
A_S1[2]="NC_SW"
A_S1[3]="W_SW"
A_S1[4]="S_SW"
A_S1[5]="SE_SW"
A_S1[6]="NE_SW"
A_S1[7]="CHT_SW"
@X&G,3 SAY "SAWN TIMBER (RWE)"
@X&G+1,3 SAY "----------------"
DO D_V_R1
RETURN
*!************************************
*!   Procedure: PGS4
*!   Calls: D_V_R1(procedure in SIM.PRG)
*!************************************
PROC PGS4
A_S1[1]="NW_PT_D"
A_S1[2]="NC_PT_D"
A_S1[3]="W_PT_D"
A_S1[4]="S_PT_D"
A_S1[5]="SE_PT_D"
A_S1[6]="NE_PT_D"
A_S1[7]="CHT_PT_D"
@X&G,3 SAY "POST & POLES
                                - DOMESTIC USE"
@X&G+1,3 SAY "-------------------------"
DO D_V_R1
RETURN
*!************************************
*!   Procedure: PGS5
*!   Calls: D_V_R1          (procedure in SIM.PRG)
*!************************************
PROC PGS5
A_S1[1]="NW_PT_I"
A_S1[2]="NC_PT_I"
A_S1[3]="W_PT_I"
A_S1[4]="S_PT_I"
A_S1[5]="SE_PT_I"
A_S1[6]="NE_PT_I"
A_S1[7]="CHT_PT_I"
@X&G,3 SAY "POST & POLES

@X&G+1,3 SAY "-------------  - INDUSTRIAL USE"
DO D_V_R1               -------------"
RETURN
*!************************************
*!   Procedure: PGS6
*!   Calls: D_V_R1
*!************************************ (procedure in SIM.PRG)
PROC PGS6
A_S1[1]="NW_PANEL"
A_S1[2]="NC_PANEL"
A_S1[3]="W_PANEL"
A_S1[4]="S_PANEL"
A_S1[5]="SE_PANEL"
A_S1[6]="NE_PANEL"
```

```
A_S1[7]="CHT_PANEL"
@X&G,3 SAY "PANEL PRODUCTS"
@X&G+1,3 SAY "--------------"
DO D_V_R1
RETURN
*!************************************
*!     Procedure: D_V_R2
*!     Called by
*!           : PGS7    (procedure in SIM.PRG)
*!           : PGS8    (procedure in SIM.PRG)
*!************************************
PROC D_V_R2
@X&G+1,20 SAY "CONSIDERING
                                   POPULATION
                     CONSIDERING POPULATION"
@X&G+2,20 SAY "GROWTH (Fixed literacy)
                         AND LITERACY GROWTH"
@X&G+3,20 SAY "('000 M.Ton)  ('000 m3)
                         ('000 M.Ton)  ('000 m3)"
@X&G+4,20 SAY "------------ -----------
                              ----------- -------"

FOR V=1 TO 2
     X=X&G+5
     SEEK A_S1[V]
     FOR I=M_I&G TO IIF(M_I&G+(18-
               X&G)*Y_I>=Y_B+Y_N,Y_B+Y_N,
                 M_I&G+(18-X&G)*Y_I) STEP Y_I
          FLD="Y"+STR(I,4,0)
          IF V=1
               @X,3 SAY "YEAR-"+STR(I,4,0)
          ENDIF
          @X,22+28*(V-1) SAY &FLD/1000
                                   PICTURE "9999.99"
          @X,34+28*(V-1) SAY &FLD/1000*
                         50/35.3 PICTURE "9999.99"
          X=X+1
     NEXT
NEXT
IF I<Y_B+Y_N+Y_I
     M_I=I
     P_CH=.F.
ELSE
     P_CH=.T.
ENDIF
RETURN
*!************************************
*!     Procedure: PGS7
*!     Calls
*!           : I_TOT   (procedure in SIM.PRG)
*!           : D_V_R2      (procedure in SIM.PRG)
*!************************************
PROC PGS7
A_S1[1]="NW_NP"
A_S1[2]="NC_NP"
A_S1[3]="W_NP"
A_S1[4]="S_NP"
A_S1[5]="SE_NP"
A_S1[6]="NE_NP"
A_S1[7]="CHT_NP"
DO I_TOT
@X&G,3 SAY "NEWSPRINT PAPER"
@X&G+1,3 SAY "---------------"
DO D_V_R2
RETURN
*!************************************
*!     Procedure: PGS8
*!     Calls
*!
*!           : I_TOT           (procedure in SIM.PRG
*!           : D_V_R2          (procedure in SIM.PRG
*!************************************
PROC PGS8
```

```
  A_S1[1]="NW_PPR"
  A_S1[2]="NC_PPR"
  A_S1[3]="W_PPR"
  A_S1[4]="S_PPR"
  A_S1[5]="SE_PPR"
  A_S1[6]="NE_PPR"
  A_S1[7]="CHT_PPR"
  DO I_TOT
  @X&G,3 SAY "OTHER PAPERS"
  @X&G+1,3 SAY "------------"
  DO D_V_R2
  RETURN
 *!*************************************
 *!   Procedure: INI_5
 *!   Called by
 *!   : S_INI         (procedure in SIM.PRG)
 *!   : S_SIM   (procedure in SIM.PRG)
 *!   : S_NF          (procedure in SIM.PRG)
 *!   : S_PL          (procedure in SIM.PRG)
 *!   : S_PR1   (procedure in SIM.PRG)
 *!*************************************
 PROC INI_5
 DECLARE A_RC[12]
 PARAMETERS ST,Z_S,A_V
 A_RC[1]=ST+"_AREA"
 A_RC[2]=ST+"_MAI"
 A_RC[3]=ST+"_AGE"
 A_RC[4]=ST+"_CRD"
 A_RC[5]=ST+"_W_C"
 A_RC[6]=ST+"_YLD"
 A_RC[7]=ST+"_TMBR"
 A_RC[8]=ST+"_POLE"
 A_RC[9]=ST+"_FUEL"
 A_RC[10]=ST+"_PULP"
 A_RC[11]=ST+"_GR_A"
 A_RC[12]=ST+"_GR_Y"
 FOR Y=1 TO 12
     SEEK A_RC[Y]
     A_V[Y]=&Z_S
 NEXT
 RETURN
*!*************************************
*!   Procedure: DB_5
*!   Called by: S_INI          (procedure in SIM.PRG)
*!*************************************
PROC DB_5
PARAMETERS ST,Z_S,A_V
DECLARE A_RC[12]
A_RC[1]=ST+"_AREA"
A_RC[2]=ST+"_MAI"
A_RC[3]=ST+"_AGE"
A_RC[4]=ST+"_CRD"
A_RC[5]=ST+"_W_C"
A_RC[6]=ST+"_YLD"
A_RC[7]=ST+"_TMBR"
A_RC[8]=ST+"_POLE"
A_RC[9]=ST+"_FUEL"
A_RC[10]=ST+"_PULP"
A_RC[11]=ST+"_GR_A"
A_RC[12]=ST+"_GR_Y"
FOR Y=1 TO 12
    SEEK A_RC[Y]
    REPL &Z_S WITH A_V[Y]
NEXT
RETURN
*!*************************************
*!   Procedure: S_INI
*!   Called by: CIRCLE           (proc in SIM.PRG)
*!   Calls
*!         : INI_5   (procedure in SIM.PRG)
*!         : DB_5    (procedure in SIM.PRG)
```

```
*!   Uses: &F_NAME1
*!   Indexes: &F_NAME1
*!   Memory Files: &F_NAME1
*!*************************************
PROC S_INI
DECLARE A_NF1[12]
DECLARE A_NF2[12]
DECLARE A_NF3[12]
DECLARE A_NF4[12]
DECLARE A_NF5[12]
DECLARE A_NF6[12]
SELE A
USE &F_NAME1 INDEX &F_NAME1
FOR XX=1 TO 6
     X=STR(XX,1,0)
     DO INI_5 WITH
                    "NF"+X,Z_STR[C1],A_NF&X
NEXT
HD="  NATURAL FOREST >  NF 1
           NF 2    NF 3    NF 4    NF 5    NF 6"
NF_C=0
HDS="   NF "
DO WHILE .T.
     @ 0,0 CLEAR TO 24, 79
     @ 0,0 TO 24, 79   DOUBLE
     @ 0,3 SAY "[ REGION: "+
                    RTRIM(SUBSTR(C[C1],6))+",
                     YEAR: "+STR(Y_B,4,0)+" ]"
     @ 2,3 SAY HD
     @ 3,3 SAY "----------------- ------- -------
                    ------- ------- ------- -------"
     @ 4,3 SAY "-Productive area(ha):"
     FOR XX=1 TO 6
          X=STR(XX,1,0)
          @4,15+9*XX GET A_NF&X[1]
                              PICTURE "9999999"
     NEXT
     READ
     @ 5,3 SAY "-MAI     (m3/ha/yr):"
     @ 6,3 SAY "-Rotation/Age  (yr):"
     @ 7,3 SAY "-Crop density      :"
     @ 8,3 SAY "-Working Cycle. (yr):"
     FOR XX=1 TO 6
          L=5
          X=STR(XX,1,0)
          FOR Y=2 TO 5
               IF A_NF&X[1]# 0
                    @L,15+9*XX GET
                      A_NF&X[Y] PICTURE "9999.99"
               ELSE
                    A_NF&X[Y]=0
               ENDIF
               L=L+1
          NEXT
     NEXT
     READ
     FOR XX=1 TO 6
          X=STR(XX,1,0)
          IF A_NF&X[1]# 0.AND.
                   A_NF&X[2]# 0.AND.A_NF&X[3]# 0
                              .AND.A_NF&X[4]# 0
               A_NF&X[6]=A_NF&X[2]*
                         A_NF&X[3]*A_NF&X[4]
          ENDIF
     NEXT
     @ 10,3 SAY "-Yield rate  (m3/ha):"
     @ 12,3 SAY "-Yield distribution"
     @ 13,3 SAY "     - Timber   (%):"
     @ 14,3 SAY "     - Pole     (%):"
     @ 15,3 SAY "     - Fuel     (%):"
     @ 16,3 SAY "     - Pulp     (%):"
@ 18,3 SAY "-Increase rate"
@ 19,3 SAY " Prod. area    (%/yr):"
@ 20,3 SAY " Yield         (%/yr):"
FOR XX=1 TO 6
    L=10
    X=STR(XX,1,0)
    FOR Y=6 TO 12
        IF A_NF&X[1]# 0
            @L,15+9*XX GET
              A_NF&X[Y] PICTURE "9999.99"
        ELSE
            A_NF&X[Y]=0
        ENDIF
        IF L# 10.AND.L# 16
            L=L+1
        ELSE
            L=L+3
        ENDIF
    NEXT
NEXT
READ
PRMPT=1
@ 24,24 PROMPT " MORE "
@ 24,32 PROMPT " NEXT "
@ 24,40 PROMPT " EDIT "
@ 24,48 PROMPT " EXIT "
MENU TO PRMPT
DO CASE
CASE PRMPT=3
    LOOP
CASE PRMPT=4
    RELEASE ALL LIKE A_*
    RETURN
ENDCASE
@24, 24 SAY " UPDATING DATA
                              FILE...WAIT... "
FOR XX=1 TO 6
    X=STR(XX,1,0)
    Y=LTRIM(STR(XX+NF_C*6,2,0))
    DO DB_5 WITH "NF"+Y,
                          Z_STR[C1],A_NF&X
NEXT
IF PRMPT=1
    IF NF_C=15
        EXIT
    ENDIF
    NF_C=NF_C+1
    IF Y_NF<=6*NF_C
        DECLARE A_RC[12]
        FOR NR=Y_NF+1 TO Y_NF+6
            Y="NF"+
                         LTRIM(STR(NR,2,0))
            A_RC[1]=Y+"_AREA"
            A_RC[2]=Y+"_MAI"
            A_RC[3]=Y+"_AGE"
            A_RC[4]=Y+"_CRD"
            A_RC[5]=Y+"_W_C"
            A_RC[6]=Y+"_YLD"
            A_RC[7]=Y+"_TMBR"
            A_RC[8]=Y+"_POLE"
            A_RC[9]=Y+"_FUEL"
            A_RC[10]=Y+"_PULP"
            A_RC[11]=Y+"_GR_A"
            A_RC[12]=Y+"_GR_Y"
            FOR X=1 TO 12
            APPEND BLANK
            REPL ITEM1 WITH
                                   A_RC[X]
            NEXT
        NEXT
        Y_NF=Y_NF+6
```

```
            SAVE ALL LIKE Y_* TO
                                 &F_NAME1
        ENDIF
        HD=" NATURAL FOREST >"
        FOR XX=1 TO 6
            X=STR(XX,1,0)
            Y=LTRIM(STR(NF_C*6+
                                 XX,2,0))
            DO INI_5 WITH "NF"+Y,
                          Z_STR[C1],A_NF&X
            HD=HD+HDS+STR(NF_C*
                                 6+XX,2,0)
        NEXT
        LOOP
    ENDIF
    EXIT
ENDDO
RELEASE ALL LIKE A_NF*
DECLARE A_PL1[12]
DECLARE A_PL2[12]
DECLARE A_PL3[12]
DECLARE A_PL4[12]
DECLARE A_PL5[12]
DECLARE A_PL6[12]
FOR XX=1 TO 6
    X=STR(XX,1,0)
    DO INI_5 WITH "PL"+X,
                          Z_STR[C1],A_PL&X
NEXT
HD="   PLANTATION   >  PL 1    PL 2
                PL 3    PL 4    PL 5    PL 6"
PL_C=0
HDS="   PL "
DO WHILE .T.
    @ 0,0  CLEAR TO 24, 79
    @ 0,0  TO 24, 79   DOUBLE
    @ 0,3 SAY "[ REGION: "+RTRIM
                (SUBSTR(C[C1],6))+", YEAR:"
                         +STR(Y_B,4,0)+"]"
    @ 2,3 SAY  HD
    @ 3,3 SAY "------------------ ------- ------
                 ------- ------- ------ -----"
    @ 4,3 SAY "-Productive area(ha):"
    FOR XX=1 TO 6
        X=STR(XX,1,0)
        @4,15+9*XX GET A_PL&X[1]
                       PICTURE "9999999"
    NEXT
    READ
    @ 5,3 SAY "-MAI        (m3/ha/yr):"
    @ 6,3 SAY "-Rotation/Age    (yr):"
    @ 7,3 SAY "-Crop density       :"
    @ 8,3 SAY "-Working Cycle. (yr):"
    FOR XX=1 TO 6
        L=5
        X=STR(XX,1,0)
        FOR Y=2 TO 5
            IF A_PL&X[1]# 0
                @L,15+9*XX GET
                  A_PL&X[Y] PICTURE "9999.9
            ELSE
                A_PL&X[Y]=0
            ENDIF
            L=L+1
        NEXT
    NEXT
    READ
    FOR XX=1 TO 6
        X=STR(XX,1,0)
        IF A_PL&X[1]# 0.AND.
              A_PL&X[2]# 0.AND.A_PL&X[3]#
```

```
                              .AND.A_PL&X[4]# 0
           A_PL&X[6]=A_PL&X[2]*
                              A_PL&X[3]*A_PL&X[4]
        ENDIF
     NEXT
     @ 10,3 SAY "-Yield rate  (m3/ha):"
     @ 12,3 SAY "-Yield distribution"
     @ 13,3 SAY "      - Timber   (%):"
     @ 14,3 SAY "      - Pole     (%):"
     @ 15,3 SAY "      - Fuel     (%):"
     @ 16,3 SAY "      - Pulp     (%):"
     @ 18,3 SAY "-Increase rate"
     @ 19,3 SAY " Prod. area   (%/yr):"
     @ 20,3 SAY " Yield        (%/yr):"
     FOR XX=1 TO 6
         L=10
         X=STR(XX,1,0)
         FOR Y=6 TO 12
             IF A_PL&X[1]# 0
                 @L,15+9*XX GET
                     A_PL&X[Y] PICTURE "9999.99"
             ELSE
                 A_PL&X[Y]=0
             ENDIF
             IF L# 10.AND.L# 16
                 L=L+1
             ELSE
                 L=L+3
             ENDIF
         NEXT
     NEXT
     READ
     PRMPT=1
     @ 24,24 PROMPT " MORE "
     @ 24,32 PROMPT " NEXT "
     @ 24,40 PROMPT " EDIT "
     @ 24,48 PROMPT " EXIT "
     MENU TO PRMPT
     DO CASE
     CASE PRMPT=3
         LOOP
     CASE PRMPT=4
         RELEASE ALL LIKE A_*
         RETURN
     ENDCASE
     @24, 24 SAY " UPDATING DATA
                               FILE...WAIT... "
     FOR XX=1 TO 6
         X=STR(XX,1,0)
         Y=LTRIM(STR(XX+PL_C*6,2,0))
         DO DB_5 WITH
                       "PL"+Y,Z_STR[C1],A_PL&X
     NEXT
     IF PRMPT=1
         IF PL_C=15
             EXIT
         ENDIF
         PL_C=PL_C+1
         IF Y_PL<=6*PL_C
             DECLARE A_RC[12]
             FOR NR=Y_PL+1 TO Y_PL+6
                 Y="PL"+
                              LTRIM(STR(NR,2,0))
                 A_RC[1]=Y+"_AREA"
                 A_RC[2]=Y+"_MAI"
                 A_RC[3]=Y+"_AGE"
                 A_RC[4]=Y+"_CRD"
                 A_RC[5]=Y+"_W_C"
                 A_RC[6]=Y+"_YLD"
                 A_RC[7]=Y+"_TMBR"
                 A_RC[8]=Y+"_POLE"
```

```
                 A_RC[9]=Y+"_FUEL"
                 A_RC[10]=Y+"_PULP"
                 A_RC[11]=Y+"_GR_A"
                 A_RC[12]=Y+"_GR_Y"
                 FOR X=1 TO 12
                     APPEND BLANK
                     REPL ITEM1 WITH
                                           A_RC[X]
                 NEXT
             NEXT
             Y_PL=Y_PL+6
             SAVE ALL LIKE Y_* TO
                                        &F_NAME1
         ENDIF
         HD="   PLANTATION   >"
         FOR XX=1 TO 6
             X=STR(XX,1,0)
             Y=LTRIM(STR(PL_C*
                                   6+XX,2,0))
             DO INI_5 WITH "PL"+Y,
                           Z_STR[C1],A_PL&X
             HD=HD+HDS+STR
                              (PL_C*6+XX,2,0)
         NEXT
         LOOP
     ENDIF
     RELEASE ALL LIKE A_PL*
     EXIT
ENDDO
RETURN
*!*************************************
*!     Procedure: VS_INI
*!     Called by: CIRCLE          (proc in SIM.PRG)
*!     Calls
*!          : INI_3   (procedure in SIM.PRG)
*!          : DB_3    (procedure in SIM.PRG)
*!     Uses: &F_NAME1
*!     Indexes: &F_NAME1
*!*************************************
PROC VS_INI
DECLARE A_V1[7]
DECLARE A_V2[7]
DECLARE A_V3[7]
DECLARE A_V4[7]
DECLARE A_V5[7] -
DECLARE A_V6[7]
DECLARE A_RC[6]
A_RC[1]="VF_TMBR"
A_RC[2]="VF_POLE"
A_RC[3]="VF_FUEL"
A_RC[4]="VF_GR_T"
A_RC[5]="VF_GR_P"
A_RC[6]="VF_GR_F"
USE &F_NAME1 INDEX &F_NAME1
FOR X=1 TO 6
    XX=LTRIM(STR(X,2,0))
    SEEK A_RC[X]
    DO INI_3 WITH A_V&XX
NEXT
DO WHILE .T.
    @ 0,0  CLEAR TO 24, 79
    @ 0,0  TO 24, 79   DOUBLE
    @ 0,5 SAY "[ INITIAL INFORMATION
              - IN THE YEAR "+STR(Y_B,4,0)+" ]"
    @ 2,3 SAY "  VILLAGE FOREST  >
                    N.WEST  N.CEN.   WEST
               SOUTH   S.EAST  N.EAST   CHT"
    @ 3,3 SAY "----------------- ------ ------
                 ------ ------ ------ ------ ------"
    @ 4,3 SAY "Yield"
    @ 6,3 SAY "-Timber volume(m3):"
```

```
   @  8,3 SAY "-Pole volume ..(m3):"
   @ 10,3 SAY "-Fuelwood vol. (m3):"
   @ 12,3 SAY "Growth rate"
   @ 14,3 SAY "   -Timber   (%/yr):"
   @ 16,3 SAY "   -Pole     (%/yr):"
   @ 18,3 SAY "   -Fuelwood (%/yr):"
   FOR X=1 TO 7
        L=6
        FOR YY=1 TO 3
             Y=LTRIM(STR(YY,2,0))
             @L,15+8*X GET A_V&Y[X]
                                  PICTURE "9999999"

             L=L+2
        NEXT
   NEXT
   FOR X=1 TO 7
        L=14
        FOR YY=4 TO 6
             Y=LTRIM(STR(YY,2,0))
             @L,15+8*X GET A_V&Y[X]
                                  PICTURE "999.999"

             L=L+2
        NEXT
   NEXT
   READ
   PRMPT=1
   @ 24,26 PROMPT " SAVE "
   @ 24,35 PROMPT " EDIT "
   @ 24,44 PROMPT " EXIT "
   MENU TO PRMPT
   DO CASE
   CASE PRMPT=2
        LOOP
   CASE PRMPT=3
        RELEASE ALL LIKE A_*
        RETURN
   ENDCASE
   @24, 26 SAY " UPDATING DATA
                                  FILE...WAIT... "

   FOR X=1 TO 6
        XX=LTRIM(STR(X,2,0))
        SEEK A_RC[X]
        DO DB_3 WITH A_V&XX
   NEXT
   EXIT
ENDDO
RETURN
*!*************************************
*!   Procedure: S_SIM
*!   Called by: SUPPLY          (proc in SIM.PRG)
*!   Calls
*!        : INI_5   (procedure in SIM.PRG)
*!        : INI_3   (procedure in SIM.PRG)
*!   Uses
*!        : &F_NAME2
*!        : &F_NAME1
*!   Indexes
*!        : &F_NAME2
*!        : &F_NAME1
*!*************************************
PROC S_SIM
@13,10 TO 17,69 DOUBLE
@15,22 SAY "SIMULATION IN
                              PROGRESS....WAIT....."
SELE B
USE &F_NAME2 INDEX &F_NAME2
SELE A
USE &F_NAME1 INDEX &F_NAME1
DECLARE A_NF1[12]
DECLARE A_NF2[12]
DECLARE A_NF3[12]
DECLARE A_NF4[12]
DECLARE A_NF5[12]
DECLARE A_NF6[12]
DECLARE A_RC2[28]
A_RC2[1]="NF_T_NW"
A_RC2[2]="NF_T_NC"
A_RC2[3]="NF_T_W"
A_RC2[4]="NF_T_S"
A_RC2[5]="NF_T_SE"
A_RC2[6]="NF_T_NE"
A_RC2[7]="NF_T_CHT"
A_RC2[8]="NF_P_NW"
A_RC2[9]="NF_P_NC"
A_RC2[10]="NF_P_W"
A_RC2[11]="NF_P_S"
A_RC2[12]="NF_P_SE"
A_RC2[13]="NF_P_NE"
A_RC2[14]="NF_P_CHT"
A_RC2[15]="NF_F_NW"
A_RC2[16]="NF_F_NC"
A_RC2[17]="NF_F_W"
A_RC2[18]="NF_F_S"
A_RC2[19]="NF_F_SE"
A_RC2[20]="NF_F_NE"
A_RC2[21]="NF_F_CHT"
A_RC2[22]="NF_PW_NW"
A_RC2[23]="NF_PW_NC"
A_RC2[24]="NF_PW_W"
A_RC2[25]="NF_PW_S"
A_RC2[26]="NF_PW_SE"
A_RC2[27]="NF_PW_NE"
A_RC2[28]="NF_PW_CHT"
DECLARE A_TY[4]
FOR X=1 TO 4
     A_TY[X]=0
NEXT
FOR V=1 TO 7
     FOR R=1 TO Y_NF/6
          SELE A
          FOR XX=1 TO 6
               X=STR(XX,1,0)
               Y=LTRIM(STR(XX+
                                        (R-1)*6,2,0))
               DO INI_5 WITH "NF"+Y,
                                   Z_STR[V],A_NF&X
          NEXT
          Y=1
          FOR I=Y_B TO Y_B+Y_N
                                        STEP Y_I
               FOR ZZ=1 TO 6
                    Z=STR(ZZ,1,0)
                    IF A_NF&Z[5]=0
                         A_NF&Z[5]=1
                    ENDIF
                    TOT=A_NF&Z[1]/
                              A_NF&Z[5]*A_NF&Z[6]*
                           (1+A_NF&Z[11]/100)^((Y-1)
                                *Y_I)*(1+A_NF&Z[12]/
                                  100)^((Y-1)*Y_I)/1000
                    A_TY[1]=A_TY[1]+TOT*
                                      A_NF&Z[7]/100
                    A_TY[2]=A_TY[2]+TOT*
                                      A_NF&Z[8]/100
                    A_TY[3]=A_TY[3]+TOT*
                                      A_NF&Z[9]/100
                    A_TY[4]=A_TY[4]+TOT*
                                      A_NF&Z[10]/100
               NEXT
               IF R=Y_NF/6
                    FLD="Y"+STR(I,4,0)
                    SELE B
```

```
                SEEK A_RC2[V]
                REPLACE &FLD
                                    WITH A_TY[1]
                SEEK A_RC2[V+7]
                REPLACE &FLD
                                    WITH A_TY[2]
                SEEK A_RC2[V+14]
                REPLACE &FLD
                                    WITH A_TY[3]
                SEEK A_RC2[V+21]
                REPLACE &FLD
                                    WITH A_TY[4]
            ENDIF
            Y=Y+1
        NEXT
    NEXT
NEXT
RELEASE ALL LIKE A_*
SELE A
DECLARE A_PL1[12]
DECLARE A_PL2[12]
DECLARE A_PL3[12]
DECLARE A_PL4[12]
DECLARE A_PL5[12]
DECLARE A_PL6[12]
DECLARE A_RC2[28]
A_RC2[1]="PL_T_NW"
A_RC2[2]="PL_T_NC"
A_RC2[3]="PL_T_W"
A_RC2[4]="PL_T_S"
A_RC2[5]="PL_T_SE"
A_RC2[6]="PL_T_NE"
A_RC2[7]="PL_T_CHT"
A_RC2[8]="PL_P_NW"
A_RC2[9]="PL_P_NC"
A_RC2[10]="PL_P_W"
A_RC2[11]="PL_P_S"
A_RC2[12]="PL_P_SE"
A_RC2[13]="PL_P_NE"
A_RC2[14]="PL_P_CHT"
A_RC2[15]="PL_F_NW"
A_RC2[16]="PL_F_NC"
A_RC2[17]="PL_F_W"
A_RC2[18]="PL_F_S"
A_RC2[19]="PL_F_SE"
A_RC2[20]="PL_F_NE"
A_RC2[21]="PL_F_CHT"
A_RC2[22]="PL_PW_NW"
A_RC2[23]="PL_PW_NC"
A_RC2[24]="PL_PW_W"
A_RC2[25]="PL_PW_S"
A_RC2[26]="PL_PW_SE"
A_RC2[27]="PL_PW_NE"
A_RC2[28]="PL_PW_CHT"
DECLARE A_TY[4]
FOR X=1 TO 4
    A_TY[X]=0
NEXT
FOR V=1 TO 7
    FOR R=1 TO Y_PL/6
        SELE A
        FOR XX=1 TO 6
            X=STR(XX,1,0)
            Y=LTRIM(STR
                                  (XX+(R-1)*6,2,0))
            DO INI_5 WITH "PL"+Y,
                                  Z_STR[V],A_PL&X
        NEXT
        Y=1
        FOR I=Y_B TO Y_B+Y_N
                                         STEP Y_I
            FOR ZZ=1 TO 6
                Z=STR(ZZ,1,0)
                IF A_PL&Z[5]=0
                    A_PL&Z[5]=1
                ENDIF
                TOT=A_PL&Z[1]/
        A_PL&Z[5]*A_PL&Z[6]*(1+A_PL&Z[11]
             /100)^((Y-1)*Y_I)*(1+A_PL&Z[12]
                          /100)^((Y-1)*Y_I)/1000
                A_TY[1]=A_TY[1]+TOT*
                                  A_PL&Z[7]/100
                A_TY[2]=A_TY[2]+TOT*
                                  A_PL&Z[8]/100
                A_TY[3]=A_TY[3]+TOT*
                                  A_PL&Z[9]/100

                A_TY[4]=A_TY[4]+TOT*
                                 A_PL&Z[10]/100
            NEXT
            IF R=Y_PL/6
                FLD="Y"+STR(I,4,0)
                SELE B
                SEEK A_RC2[V]
                REPLACE &FLD
                                    WITH A_TY[1]
                SEEK A_RC2[V+7]
                REPLACE &FLD
                                    WITH A_TY[2]
                SEEK A_RC2[V+14]
                REPLACE &FLD
                                    WITH A_TY[3]
                SEEK A_RC2[V+21]
                REPLACE &FLD
                                    WITH A_TY[4]
            ENDIF
            Y=Y+1
        NEXT
    NEXT
NEXT
RELEASE ALL LIKE A_*
SELE A
DECLARE A_V1[7]
DECLARE A_V2[7]
DECLARE A_V3[7]
DECLARE A_V4[7]
DECLARE A_V5[7]
DECLARE A_V6[7]
DECLARE A_RC1[6]
A_RC1[1]="VL_TMBR"
A_RC1[2]="VF_POLE"
A_RC1[3]="VF_FUEL"
A_RC1[4]="VF_GR_T"
A_RC1[5]="VF_GR_P"
A_RC1[6]="VF_GR_F"
FOR X=1 TO 6
    XX=LTRIM(STR(X,2,0))
    SEEK A_RC1[X]
    DO INI_3 WITH A_V&XX
NEXT
DECLARE A_RC2[21]
A_RC2[1]="VF_T_NW"
A_RC2[2]="VF_T_NC"
A_RC2[3]="VF_T_W"
A_RC2[4]="VF_T_S"
A_RC2[5]="VF_T_SE"
A_RC2[6]="VF_T_NE"
A_RC2[7]="VF_T_CHT"
A_RC2[8]="VF_P_NW"
A_RC2[9]="VF_P_NC"
A_RC2[10]="VF_P_W"
A_RC2[11]="VF_P_S"
```

```
A_RC2[12]="VF_P_SE"
A_RC2[13]="VF_P_NE"
A_RC2[14]="VF_P_CHT"
A_RC2[15]="VF_F_NW"
A_RC2[16]="VF_F_NC"
A_RC2[17]="VF_F_W"
A_RC2[18]="VF_F_S"
A_RC2[19]="VF_F_SE"
A_RC2[20]="VF_F_NE"
A_RC2[21]="VF_F_CHT"
DECLARE A_TY[7]
SELE B
FOR V=1 TO 7
     Y=1
     FOR I=Y_B TO Y_B+Y_N STEP Y_I
          FLD="Y"+STR(I,4,0)
          A_TY[V]=A_V1[V]*
                    (1+A_V4[V]/100)^((Y-1)*Y_I)/1000
          SEEK A_RC2[V]
          REPLACE &FLD WITH A_TY[V]
          A_TY[V]=A_V2[V]*
                    (1+A_V5[V]/100)^((Y-1)*Y_I)/1000
          SEEK A_RC2[V+7]
          REPLACE &FLD WITH A_TY[V]
          A_TY[V]=A_V3[V]*
                    (1+A_V6[V]/100)^((Y-1)*Y_I)/1000
          SEEK A_RC2[V+14]
          REPLACE &FLD WITH A_TY[V]
          Y=Y+1
     NEXT
NEXT
RELEASE  ALL LIKE A_*
RETURN
*!**************************************
*!    Procedure: S_VW
*!    Called by: SUPPLY              (proc in SIM.PRG)
*!    Calls
*!         : Z_NOTE(procedure in SIM.PRG)
*!         : S_VW1  (procedure in SIM.PRG)
*!         : S_VW2  (procedure in SIM.PRG)
*!**************************************
PROC S_VW
DO WHILE S2#0
     @9,47 TO 12,68 DOUBLE
     @10,48 PROMPT " INITIAL
                                   PARAMETERS "
     @11,48 PROMPT " SIMULATION
                                      RESULTS "
     MENU TO S2
     IF S2#0
          SAVE SCREEN TO SCR3
          DO CASE
          CASE S2=1
               DO Z_NOTE
               DO S_VW1
          CASE S2=2
               DO S_VW2
          ENDCASE
          CLOSE DATABASES
          REST SCREEN FROM SCR3
     ENDIF
ENDDO
RETURN
*!**************************************
*!    Procedure: S_VW1
*!    Called by: S_VW
*!    Calls                         (procedure in SIM.PRG)
*!         : INI_3    (procedure in SIM.PRG)
*!         : S_NF     (procedure in SIM.PRG)
*!         : S_PL     (procedure in SIM.PRG)
*!         : SPAGE15
                                    (procedure in SIM.PRG)
```

```
*!    Uses: &F_NAME1
*!    Indexes: &F_NAME1
*!**************************************
PROC S_VW1
SELE A
USE &F_NAME1 INDEX &F_NAME1
DECLARE A_NF1[12]
DECLARE A_NF2[12]
DECLARE A_NF3[12]
DECLARE A_NF4[12]
DECLARE A_NF5[12]
DECLARE A_NF6[12]
DECLARE A_PL1[12]
DECLARE A_PL2[12]
DECLARE A_PL3[12]
DECLARE A_PL4[12]
DECLARE A_PL5[12]
DECLARE A_PL6[12]
DECLARE A_V1[7]
DECLARE A_V2[7]
DECLARE A_V3[7]
DECLARE A_V4[7]
DECLARE A_V5[7]
DECLARE A_V6[7]
DECLARE A_RC[6]
A_RC[1]="VF_TMBR"
A_RC[2]="VF_POLE"
A_RC[3]="VF_FUEL"
A_RC[4]="VF_GR_T"
A_RC[5]="VF_GR_P"
A_RC[6]="VF_GR_F"
USE &F_NAME1 INDEX &F_NAME1
FOR X=1 TO 6
     XX=LTRIM(STR(X,2,0))
     SEEK A_RC[X]
     DO INI_3 WITH A_V&XX
NEXT
@ 0,0 CLEAR TO 24,79
@ 0,0 TO 24,79 DOUBLE
@ 0,72 SAY "PgUp-"+CHR(24)
@ 24,72 SAY "PgDn-"+CHR(25)
@ 24,2 SAY "Esc-exit"
@0,20 SAY ' INITIAL PARAMETERS FOR
                              SUPPLY SIMULATION'
LX=1
P_CH=1
ST=0
PG=1
I_P=0
I_P1=Y_B
V1=2
G=LTRIM(STR(PG,2,0))
V=0
DO WHILE .T.
     @ 0,5 TO 0,15 DOUBLE
     @ 0,5 SAY "Page "+G
     DO CASE
     CASE Y_NF/6*7>=P_CH
          DO CASE
          CASE P_CH<=1*Y_NF/6
               DO S_NF
                              WITH Z_STR[1],P_CH,1
          CASE P_CH<=2*Y_NF/6
               DO S_NF WITH
                         Z_STR[2],P_CH-Y_NF/6*1,2
          CASE P_CH<=3*Y_NF/6
               DO S_NF WITH
                         Z_STR[3],P_CH-Y_NF/6*2,3
          CASE P_CH<=4*Y_NF/6
               DO S_NF WITH
                         Z_STR[4],P_CH-Y_NF/6*3,4
```

```
        CASE P_CH <=5*Y_NF/6
            DO S_NF WITH
                        Z_STR[5],P_CH-Y_NF/6*4,5
        CASE P_CH <=6*Y_NF/6
            DO S_NF WITH
                        Z_STR[6],P_CH-Y_NF/6*5,6
        CASE P_CH <=7*Y_NF/6
            DO S_NF WITH
                        Z_STR[7],P_CH-Y_NF/6*6,7
        ENDCASE
    CASE Y_NF/6*7+Y_PL/6*7>=P_CH
        P_PL=P_CH-Y_NF/6*7
        DO CASE
        CASE P_PL<=1*Y_PL/6
            DO S_PL WITH
                                    Z_STR[1],P_PL,1
        CASE P_PL<=2*Y_PL/6
            DO S_PL WITH
                        Z_STR[2],P_PL-Y_PL/6*1,2
        CASE P_PL<=3*Y_PL/6
            DO S_PL WITH
                        Z_STR[3],P_PL-Y_PL/6*2,3
        CASE P_PL<=4*Y_PL/6
            DO S_PL WITH
                        Z_STR[4],P_PL-Y_PL/6*3,4
        CASE P_PL<=5*Y_PL/6
            DO S_PL WITH
                        Z_STR[5],P_PL-Y_PL/6*4,5
        CASE P_PL<=6*Y_PL/6
            DO S_PL WITH
                        Z_STR[6],P_PL-Y_PL/6*5,6
        CASE P_PL<=7*Y_PL/6
            DO S_PL WITH
                        Z_STR[7],P_PL-Y_PL/6*6,7
        ENDCASE
    OTHERWISE
        DO SPAGE15
    ENDCASE
    KEY=0
    DO WHILE .T.
        KEY=INKEY()
        DO CASE
        CASE KEY=27
            RETURN
        CASE KEY=3.OR.KEY=18
            EXIT
        ENDCASE
    ENDDO
    DO CASE
    CASE KEY=18
        PG=PG-1
        P_CH=P_CH-1
    CASE KEY=3
        PG=PG+1
        P_CH=P_CH+1
    ENDCASE
    G=LTRIM(STR(PG,2,0))
    IF PG=0.OR.P_CH=0.OR.P_CH=
                        (Y_NF/6+Y_PL/6)*7+2
        RETURN
    ENDIF
    @1,1 CLEAR TO 23,78
ENDDO
RELEASE ALL LIKE A_*
RETURN
*!*************************************
*!    Procedure: S_NF
*!    Called by: S_VW1         (procedure in SIM.PRG)
*!    Calls: INI_5             (procedure in SIM.PRG)
*!*************************************
PROC S_NF
```

```
PARAMETERS A_Z,NF_C,R_N
HD=" NATURAL FOREST >"
HDS="   NF "
FOR XX=1 TO 6
    X=STR(XX,1,0)
    Y=LTRIM(STR((NF_C-1)*6+XX,2,0))
    DO INI_5 WITH "NF"+Y,A_Z,A_NF&X
    HD=HD+HDS+STR((NF_C-1)*
                                  6+XX,2,0)
NEXT
@ 2,3 SAY "[ REGION: "+
                    RTRIM(SUBSTR(C[R_N],6))+",
                      YEAR: "+STR(Y_B,4,0)+" ]"
@2+ 2,3 SAY HD
@2+ 3,3 SAY "------------------ ------- -------
                    ------- ------- ------- -------"
@2+ 4,3 SAY "-Productive area(ha):"
@2+ 5,3 SAY "-MAI     (m3/ha/yr):"
@2+ 6,3 SAY "-Age          (yr):"
@2+ 7,3 SAY "-Crop density     :"
@2+ 8,3 SAY "-Working Cycle. (yr):"
@2+ 10,3 SAY "-Yield rate (m3/ha):"
@2+ 12,3 SAY "-Yield distribution"
@2+ 13,3 SAY "     - Timber   (%):"
@2+ 14,3 SAY "     - Pole     (%):"
@2+ 15,3 SAY "     - Fuel     (%):"
@2+ 16,3 SAY "     - Pulp     (%):"
@2+ 18,3 SAY "-Increase rate"
@2+ 19,3 SAY " Prod. area   (%/yr):"
@2+ 20,3 SAY " Yield        (%/yr):"
FOR XX=1 TO 6
    X=STR(XX,1,0)
    @2+4,15+9*XX SAY A_NF&X[1]
                            PICTURE "9999999"
    L=5
    FOR Y=2 TO 5
        IF A_NF&X[1]#0
            @2+L,15+9*XX SAY
                    A_NF&X[Y] PICTURE "999.999"
        ENDIF
        L=L+1
    NEXT
NEXT
FOR XX=1 TO 6
    L=10
    X=STR(XX,1,0)
    FOR Y=6 TO 12
        IF A_NF&X[1]#0
            @2+L,15+9*XX SAY
                    A_NF&X[Y] PICTURE "999.999"
        ENDIF
        IF L#10.AND.L#16
            L=L+1
        ELSE
            L=L+3
        ENDIF
    NEXT
NEXT
RETURN
*!*************************************
*!    Procedure: S_PL
*!    Called by: S_VW1         (procedure in SIM.PRG)
*!    Calls: INI_5             (procedure in SIM.PRG)
*!*************************************
PROC S_PL
PARAMETERS A_Z, PL_C, R_N
HD="   PLANTATION    >"
HDS="   PL "
FOR XX=1 TO 6
    X=STR(XX,1,0)
    Y=LTRIM(STR((PL_C-1)*6+XX,2,0))
    DO INI_5 WITH "PL"+Y,A_Z,A_PL&X
    HD=HD+HDS+STR((PL_C-1)*
                                        6+XX,2,0)
NEXT
@ 2,3 SAY "[ REGION: "+RTRIM
                             (SUBSTR(C[R_N],6))+",
                         YEAR: "+STR(Y_B,4,0)+" ]"
@2+  2,3 SAY HD
@2+  3,3 SAY "----------------- ------- -------
                              ------- ------- ------- -------"
@2+  4,3 SAY "-Productive area(ha):"
@2+  5,3 SAY "-MAI       (m3/ha/yr):"
@2+  6,3 SAY "-Age             (yr):"
@2+  7,3 SAY "-Crop density        :"
@2+  8,3 SAY "-Working Cycle. (yr):"
@2+ 10,3 SAY "-Yield rate  (m3/ha):"
@2+ 12,3 SAY "-Yield distribution"
@2+ 13,3 SAY "      - Timber   (%):"
@2+ 14,3 SAY "      - Pole     (%):"
@2+ 15,3 SAY "      - Fuel     (%):"
@2+ 16,3 SAY "      - Pulp     (%):"
@2+ 18,3 SAY "-Increase rate"
@2+ 19,3 SAY " Prod. area   (%/yr):"
@2+ 20,3 SAY " Yield        (%/yr):"
FOR XX=1 TO 6
    X=STR(XX,1,0)
    @2+4,15+9*XX   SAY   A_PL&X[1]   PICTURE
"9999999"
NEXT
FOR XX=1 TO 6
    L=5
    X=STR(XX,1,0)
    FOR Y=2 TO 5
        IF A_PL&X[1]# 0
            @2+L,15+9*XX SAY
                     A_PL&X[Y] PICTURE "999.999"
        ENDIF
        L=L+1
    NEXT
NEXT
FOR XX=1 TO 6
    L=10
    X=STR(XX,1,0)
    FOR Y=6 TO 12
        IF A_PL&X[1]# 0
            @2+L,15+9*XX SAY
                     A_PL&X[Y] PICTURE "999.999"
        ENDIF
        IF L# 10.AND.L# 16
            L=L+1
        ELSE
            L=L+3
        ENDIF
    NEXT
NEXT
RETURN
*!**************************************
*!   Procedure: SPAGE15
*!   Called by: S_VW1              (proc in SIM.PRG)
*!**************************************
PROC SPAGE15
@2+ 0,5 SAY "[ INITIAL INFORMATION:
                          YEAR "+STR(Y_B,4,0)+" ]"
@2+ 2,3 SAY "< VILLAGE FOREST  >
              N.WEST  N.CEN.  WEST  SOUTH
                          S.EAST  N.EAST  CHT"
@2+ 3,3 SAY "----------------- ------ -------
                    ------- ------- ------- ------- -------"
@2+ 4,3 SAY "Yield"
@2+ 6,3 SAY "-Timber volume.(m3):"
@2+ 8,3 SAY "-Pole volume ..(m3):"
@2+ 10,3 SAY "-Fuelwood vol. (m3):"
@2+ 12,3 SAY "Growth rate"
@2+ 14,3 SAY "  -Timber    (%/yr):"
@2+ 16,3 SAY "  -Pole      (%/yr):"
@2+ 18,3 SAY "  -Fuelwood (%/yr):"
FOR X=1 TO 7
    L=6
    FOR YY=1 TO 3
        Y=LTRIM(STR(YY,2,0))
        @2+L,15+8*X SAY A_V&Y[X]
                              PICTURE "99999.9"

        L=L+2
    NEXT
NEXT
FOR X=1 TO 7
    L=14
    FOR YY=4 TO 6
        Y=LTRIM(STR(YY,2,0))
        @2+L,15+8*X SAY A_V&Y[X]
                              PICTURE "999.999"

        L=L+2
    NEXT
NEXT
RETURN
*!**************************************
*!     Procedure: S_VW2
*!     Called by: S_VW          (procedure in SIM.PRG)
*!     Calls: &PROX
*!     Uses: &F_NAME2
*!     Indexes: &F_NAME2
*!**************************************
PROC S_VW2
SELE A
USE &F_NAME2 INDEX &F_NAME2
@ 0,0 CLEAR TO 24,79
@ 0,0 TO 24,79 DOUBLE
@ 0,72 SAY "PgUp-"+CHR(24)
@ 24,72 SAY "PgDn-"+CHR(25)
@ 24,2 SAY "Esc-exit"
@0,20 SAY ' FUTURE PROJECTIONS
                    FROM SUPPLY SIMULATION'
HD1="          N.WEST N.CENTR.
                       WEST   SOUTH   S.EAST
                    N.EAST   CHT   NATIONAL'
HD2="          ------- ------- ------- -------
                    ------- ------- ------- -------'
LX=1
ST=0
PG=1
P_CH=.T.
PGS=1
PGN=1
M_I1=Y_B
M_I=Y_B
G='1'
DECLARE A_RC2[7]
DECLARE TOT[Y_N+1]
FOR Z=1 TO Y_N+1
    TOT[Z]=0
NEXT
I=0
X1=2
X=5
PGS1=1
T='1'
CH_PGN=.T.
PGS_E=12
DO WHILE .T.
    @0,5 TO 0,15 DOUBLE
    @0,5 SAY "Page "+T
    PROX='SPGS'+LTRIM(STR(PGS,2,0))
```

```
      DO &PROX
      IF X>18.OR.PGS=PGS_E-1
          KEY=0
          DO WHILE .T.
              KEY=INKEY()
              DO CASE
              CASE KEY=27
                  RETURN
              CASE KEY=3.OR.KEY=18
                  EXIT
              ENDCASE
          ENDDO
     ELSE
         KEY=3
     ENDIF
     DO CASE
     CASE KEY=18
         IF !CH_PGN
              DO  WHILE X&G#2
                  PG=PG-1
                  G=LTRIM(STR(PG,2,0))
              ENDDO
         ENDIF
         PG=PG-1
         G=LTRIM(STR(PG,2,0))
         IF PG=0
              RETURN
         ENDIF
         DO WHILE X&G#2
              PG=PG-1
              G=LTRIM(STR(PG,2,0))
         ENDDO
         PGN=PGN-1
         T=LTRIM(STR(PGN,2,0))
         PGS=PGS&T
     CASE KEY=3
         PG=PG+1
         G=LTRIM(STR(PG,2,0))
         IF P_CH
              FOR Z=1 TO Y_N+1
                  TOT[Z]=0
              NEXT
              PGS=PGS+1
              IF PGS_E=16.AND.PGS=8
                  PGS=9
              ENDIF
              M_I&G=Y_B
              M_I=Y_B
         ELSE
              X=X+1
              M_I&G=M_I
         ENDIF
         IF X>18
              X&G=2
              X=5
              PGN=PGN+1
              T=LTRIM(STR(PGN,2,0))
              PGS&T=PGS
              CH_PGN=.T.
         ELSE
              X=X+1
              X&G=X
              CH_PGN=.F.
         ENDIF
     ENDCASE
     IF PG=0.OR.PGS=PGS_E
         RETURN
     ENDIF
     @X&G,1 CLEAR TO 23,78
ENDDO
RETURN
```

```
*!**************************************
*!     Procedure: SPGS1
*!     Calls: S_V_R1          (procedure in SIM.PRG)
*!**************************************
PROC SPGS1
A_RC2[1]="NF_T_NW"
A_RC2[2]="NF_T_NC"
A_RC2[3]="NF_T_W"
A_RC2[4]="NF_T_S"
A_RC2[5]="NF_T_SE"
A_RC2[6]="NF_T_NE"
A_RC2[7]="NF_T_CHT"
@X&G,3 SAY    "TIMBER - NATURAL FOREST"
@X&G+1,3 SAY  "-----------------------"
DO S_V_R1
RETURN
*!**************************************
*!     Procedure: S_V_R1
*!     Called by
*!     : SPGS1  (procedure in SIM.PRG)
*!     : SPGS2  (procedure in SIM.PRG)
*!     : SPGS3  (procedure in SIM.PRG)
*!     : SPGS4  (procedure in SIM.PRG)
*!     : SPGS5  (procedure in SIM.PRG)
*!     : SPGS6  (procedure in SIM.PRG)
*!     : SPGS7  (procedure in SIM.PRG)
*!     : SPGS8  (procedure in SIM.PRG)
*!     : SPGS9  (procedure in SIM.PRG)
*!     : SPGS10 (procedure in SIM.PRG)
*!     : SPGS11 (procedure in SIM.PRG)
*!**************************************
PROC S_V_R1
@X&G,66 SAY "(in '000 m3)"
@X&G+2,3 SAY HD1
@X&G+3,3 SAY HD2
FOR V=1 TO 8
     X=X&G+4
     IF V#8
          SEEK A_RC2[V]
     ENDIF
     FOR I=M_I&G TO IIF(M_I&G+
             (18-X&G)*Y_I>=Y_B+Y_N,Y_B+Y_N,
                M_I&G+(18-X&G)*Y_I) STEP Y_I
          IF V#8
              FLD="Y"+STR(I,4,0)
              IF V=1
                  @X,3 SAY
                              "YEAR-"+STR(I,4,0)
              ENDIF
              @X,6+8*V SAY &FLD
                               PICTURE "9999.99"
              TOT[(I-Y_B)/Y_I+1]=
                      TOT[(I-Y_B)/Y_I+1]+&FLD
          ELSE
              @X,6+8*V SAY
           TOT[(I-Y_B)/Y_I+1] PICTURE "99999.99"
          ENDIF
          X=X+1
     NEXT
NEXT
IF I<Y_B+Y_N+Y_I
     M_I=I
     P_CH=.F.
ELSE
     P_CH=.T.
ENDIF
RETURN
*!**************************************
*!     Procedure: SPGS2
*!     Calls: S_V_R1          (procedure in SIM.PRG)
*!**************************************
```

```
PROC SPGS2
A_RC2[1]="PL_T_NW"
A_RC2[2]="PL_T_NC"
A_RC2[3]="PL_T_W"
A_RC2[4]="PL_T_S"
A_RC2[5]="PL_T_SE"
A_RC2[6]="PL_T_NE"
A_RC2[7]="PL_T_CHT"
@X&G,3 SAY   "TIMBER - PLANTATION"
@X&G+1,3 SAY  "------------------"
DO S_V_R1
RETURN
*!*************************************
*!    Procedure: SPGS3
*!    Calls: S_V_R1          (procedure in SIM.PRG)
*!*************************************
PROC SPGS3
A_RC2[1]="VF_T_NW"
A_RC2[2]="VF_T_NC"
A_RC2[3]="VF_T_W"
A_RC2[4]="VF_T_S"
A_RC2[5]="VF_T_SE"
A_RC2[6]="VF_T_NE"
A_RC2[7]="VF_T_CHT"
@X&G,3 SAY   "TIMBER - VILLAGE       FOREST"
@X&G+1,3 SAY  "---------------------"
DO S_V_R1
RETURN
*!*************************************
*!    Procedure: SPGS4
*!    Calls: S_V_R1          (procedure in SIM.PRG)
*!*************************************
PROC SPGS4
A_RC2[1]="NF_P_NW"
A_RC2[2]="NF_P_NC"
A_RC2[3]="NF_P_W"
A_RC2[4]="NF_P_S"
A_RC2[5]="NF_P_SE"
A_RC2[6]="NF_P_NE"
A_RC2[7]="NF_P_CHT"
@X&G,3 SAY   "POLE - NATURAL

                                         FOREST"
@X&G+1,3 SAY  "-------------------"
DO S_V_R1
RETURN
*!**********************************************
*******************
*!    Procedure: SPGS5
*!    Calls: S_V_R1(          procedure in SIM.PRG)
*!*************************************
PROC SPGS5
A_RC2[1]="PL_P_NW"
A_RC2[2]="PL_P_NC"
A_RC2[3]="PL_P_W"
A_RC2[4]="PL_P_S"
A_RC2[5]="PL_P_SE"
A_RC2[6]="PL_P_NE"
A_RC2[7]="PL_P_CHT"
@X&G,3 SAY   "POLE - PLANTATION"
@X&G+1,3 SAY  "----------------"
DO S_V_R1
RETURN
*!*************************************
*!    Procedure: SPGS6
*!    Calls: S_V_R1
*!*************************************          (procedure in SIM.PRG)
PROC SPGS6
A_RC2[1]="VF_P_NW"
A_RC2[2]="VF_P_NC"
A_RC2[3]="VF_P_W"
A_RC2[4]="VF_P_S"
```

(Proj. 372001/25, App. 3)

```
A_RC2[5]="VF_P_SE"
A_RC2[6]="VF_P_NE"
A_RC2[7]="VF_P_CHT"
@X&G,3 SAY   "POLE - VILLAGE
                                         FOREST"
@X&G+1,3 SAY  "-------------------"
DO S_V_R1
RETURN
*!*************************************
*!    Procedure: SPGS7
*!    Calls: S_V_R1          (procedure in SIM.PRG)
*!*************************************
PROC SPGS7
A_RC2[1]="NF_F_NW"
A_RC2[2]="NF_F_NC"
A_RC2[3]="NF_F_W"
A_RC2[4]="NF_F_S"
A_RC2[5]="NF_F_SE"
A_RC2[6]="NF_F_NE"
A_RC2[7]="NF_F_CHT"
@X&G,3 SAY   "FUELWOOD - NATURAL
                                         FOREST"
@X&G+1,3 SAY  "----------------------"
DO S_V_R1
RETURN
*!*************************************
*!    Procedure: SPGS8
*!    Calls: S_V_R1          (procedure in SIM.PRG)
*!*************************************
PROC SPGS8
A_RC2[1]="PL_F_NW"
A_RC2[2]="PL_F_NC"
A_RC2[3]="PL_F_W"
A_RC2[4]="PL_F_S"
A_RC2[5]="PL_F_SE"
A_RC2[6]="PL_F_NE"
A_RC2[7]="PL_F_CHT"
@X&G,3 SAY   "FUELWOOD -

                                     PLANTATION"
@X&G+1,3 SAY  "--------------------"
DO S_V_R1
RETURN
*!*************************************
*!    Procedure: SPGS9
*!    Calls: S_V_R1
*!*************************************     (procedure in SIM.PRG)
PROC SPGS9
A_RC2[1]="VF_F_NW"
A_RC2[2]="VF_F_NC"
A_RC2[3]="VF_F_W"
A_RC2[4]="VF_F_S"
A_RC2[5]="VF_F_SE"
A_RC2[6]="VF_F_NE"
A_RC2[7]="VF_F_CHT"
@X&G,3 SAY   "FUELWOOD - VILLAGE
@X&G+1,3 SAY  "-----------------------"      FOREST"
DO S_V_R1
RETURN

*!*************************************
*!    Procedure: SPGS10
*!    Calls: S_V_R1
*!*************************************     (procedure in SIM.PRG)
PROC SPGS10
A_RC2[1]="NF_PW_NW"
A_RC2[2]="NF_PW_NC"
A_RC2[3]="NF_PW_W"
A_RC2[4]="NF_PW_S"
A_RC2[5]="NF_PW_SE"
A_RC2[6]="NF_PW_NE"
```

```
A_RC2[7]="NF_PW_CHT"
@X&G,3 SAY   "PULPWOOD - NATURAL
                                        FOREST"
@X&G+1,3 SAY  "------------------------"
DO S_V_R1
RETURN
*!*************************************
*!   Procedure: SPGS11
*!   Calls: S_V_R1          (procedure in SIM.PRG)
*!*************************************
PROC SPGS11
A_RC2[1]="PL_PW_NW"
A_RC2[2]="PL_PW_NC"
A_RC2[3]="PL_PW_W"
A_RC2[4]="PL_PW_S"
A_RC2[5]="PL_PW_SE"
A_RC2[6]="PL_PW_NE"
A_RC2[7]="PL_PW_CHT"
@X&G,3 SAY   "PULPWOOD -
                                   PLANTATION"
@X&G+1,3 SAY  "--------------------"
DO S_V_R1
RETURN
*!*************************************
*!   Procedure: Z_NOTE
*!   Called by
*!        : D_VW           (procedure in SIM.PRG)
*!        : S_VW           (procedure in SIM.PRG)
*!*************************************
PROC Z_NOTE
@ 6,10 CLEAR TO 21,69
@ 6,10 TO 21,69 DOUBLE
@ 7,33 SAY "ZONE DEFINITION"
@ 8,33 SAY "---------------"
@9,13 SAY "NORTH WEST    - Districts of
                          Dinajpur, Rangpur, Bogra,"
@10,13 SAY "            Rajshahi and Pabna"
@11,13 SAY "NORTH CENTRAL - Districts
                         of Dhaka, Tangail, Jamalpur"
@12,13 SAY "            and
                                  Mymensingh(part)"
@13,13 SAY "WEST          - Districts of
                           Kushtia, Jessore, Barisal,"
@14,13 SAY "            Faridpur and
                                      Khulna(part)"
@15,13 SAY "SOUTH         - Districts of
                           Khulna(part), Patuakhali"
@16,13 SAY "SOUTH EAST    - Districts of
                            Comilla(part), Noakhali"
@17,13 SAY "            and Chittagong"
@18,13 SAY "NORTH EAST    - Districts of
                         Mymnesingh(part), Sylhet"
@19,13 SAY "            and Comilla(part)"
@20,13 SAY "CHT           - District of
                              Chittagong Hill Tracts"
@21,13 SAY "[ Press any key to continue....]"
INKEY(0)
RETURN
*!*************************************
*!   Procedure: B_VW
*!   Called by: BALANCE          (proc in SIM.PRG)
*!   Calls: &PROX
*!   Uses: &F_NAME2
*!   Indexes: &F_NAME2
*!*************************************
PROC B_VW
SELE A
USE &F_NAME2 INDEX &F_NAME2
@ 0,0 CLEAR TO 24,79
@ 0,0 TO 24,79 DOUBLE
@ 0,72 SAY "PgUp-"+CHR(24)
```

```
@ 24,72 SAY "PgDn-"+CHR(25)
@ 24,2 SAY "Esc-exit"
@0,20 SAY ' BALANCE FROM FUTURE
                                  PROJECTIONS '
HD1="           N.WEST N.CENTR.
                      WEST  SOUTH  S.EAST
                  N.EAST  CHT  NATIONAL"
HD2="         ------- ------- ------- -------
                   ------- ------- ------- --------"
LX=1
ST=0
PG=1
P_CH=.T.
PGS=1
PGN=1
M_I1=Y_B
M_I=Y_B
G='1'
DECLARE A_S1[7]
DECLARE A_S2[7]
DECLARE A_S3[7]
DECLARE A_S4[7]
DECLARE A_S5[7]
DECLARE A_D_T[8]
DECLARE A_S_T[8]
DECLARE A_DOT[Y_N+1]
DECLARE A_INT[Y_N+1]
DECLARE A_NFT[Y_N+1]
DECLARE A_PLT[Y_N+1]
DECLARE A_VFT[Y_N+1]
X1=2
X=5
PGS1=1
T='1'
CH_PGN=.T.
PGS_E=6
DO WHILE .T.
    @0,5 TO 0,15 DOUBLE
    @0,5 SAY "Page "+T
    PROX='BPGS'+LTRIM(STR(PGS,2,0))
    DO &PROX
    IF X>18.OR.PGS=PGS_E-1
        KEY=0
        DO WHILE .T.
            KEY=INKEY()
            DO CASE
            CASE KEY=27
                RETURN
            CASE KEY=3.OR.KEY=18
                EXIT
            ENDCASE
        ENDDO
    ELSE
        KEY=3
    ENDIF
    DO CASE
    CASE KEY=18
        IF !CH_PGN
            DO WHILE X&G#2
                PG=PG-1
                G=LTRIM(STR(PG,2,0))
            ENDDO
        ENDIF
        PG=PG-1
        G=LTRIM(STR(PG,2,0))
        IF PG=0
            RETURN
        ENDIF
        DO WHILE X&G#2
            PG=PG-1
            G=LTRIM(STR(PG,2,0))
        ENDDO
        PGN=PGN-1
        T=LTRIM(STR(PGN,2,0))
        PGS=PGS&T
    CASE KEY=3
        PG=PG+1
        G=LTRIM(STR(PG,2,0))
        IF P_CH
            PGS=PGS+1
            IF PGS_E=16.AND.PGS=8
                PGS=9
            ENDIF
            M_I&G=Y_B
            M_I=Y_B
        ELSE
            X=X+1
            M_I&G=M_I
        ENDIF
        IF X>18
            X&G=2
            X=5
            PGN=PGN+1
            T=LTRIM(STR(PGN,2,0))
            PGS&T=PGS
            CH_PGN=.T.
        ELSE
            X=X+1
            X&G=X
            CH_PGN=.F.
        ENDIF
    ENDCASE
    IF PG=0.OR.PGS=PGS_E
        RETURN
    ENDIF
    @X&G,1 CLEAR TO 23,78
ENDDO
RETURN
*!**************************************
*!    Procedure: I_T8
*!    Called by
*!          : BPGS1  (proc in SIM.PRG)
*!          : BPGS2  (procedure in SIM.PRG)
*!          : BPGS3  (procedure in SIM.PRG)
*!          : BPGS4  (procedure in SIM.PRG)
*!          : BPGS5  (procedure in SIM.PRG)
*!          : B_PR   (procedure in SIM.PRG)
*!**************************************
PROC I_T8
PARAMETERS A_TX
FOR Z=1 TO 8
    A_TX[Z]=0
NEXT
RETURN

*!**************************************
*!     Procedure: I_TY
*!          : BPGS1  (proc in SIM.PRG)
*!          : BPGS2  (procedure in SIM.PRG)
*!          : BPGS3  (procedure in SIM.PRG)
*!          : BPGS4  (procedure in SIM.PRG)
*!          : BPGS5  (procedure in SIM.PRG)
*!          : B_PR   (procedure in SIM.PRG)
*!**************************************

PROC I_TY
PARAMETERS A_TX
FOR Z=1 TO Y_N+1
    A_TX[Z]=0
NEXT
RETURN
*!**************************************
```

```
*!     Procedure: BPGS1
*!     Calls:
*!     : I_T8      (procedure in SIM.PRG)
*!     : I_TY      (procedure in SIM.PRG)
*!**************************************
PROC BPGS1
A_S1[1]="NW_FWD_D"
A_S1[2]="NC_FWD_D"
A_S1[3]="W_FWD_D"
A_S1[4]="S_FWD_D"
A_S1[5]="SE_FWD_D"
A_S1[6]="NE_FWD_D"
A_S1[7]="CHT_FWD_D"
A_S2[1]="NW_FWD_I"
A_S2[2]="NC_FWD_I"
A_S2[3]="W_FWD_I"
A_S2[4]="S_FWD_I"
A_S2[5]="SE_FWD_I"
A_S2[6]="NE_FWD_I"
A_S2[7]="CHT_FWD_I"
A_S3[1]="NF_F_NW"
A_S3[2]="NF_F_NC"
A_S3[3]="NF_F_W"
A_S3[4]="NF_F_S"
A_S3[5]="NF_F_SE"
A_S3[6]="NF_F_NE"
A_S3[7]="NF_F_CHT"
A_S4[1]="PL_F_NW"
A_S4[2]="PL_F_NC"
A_S4[3]="PL_F_W"
A_S4[4]="PL_F_S"
A_S4[5]="PL_F_SE"
A_S4[6]="PL_F_NE"
A_S4[7]="PL_F_CHT"
A_S5[1]="VF_F_NW"
A_S5[2]="VF_F_NC"
A_S5[3]="VF_F_W"
A_S5[4]="VF_F_S"
A_S5[5]="VF_F_SE"
A_S5[6]="VF_F_NE"
A_S5[7]="VF_F_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY   "FUELWOOD"
@3,3 SAY  "--------"
@3,66 SAY "(in '000 m3)"
@4,3 SAY HD1
@5,3 SAY HD2
I=M_I&G
FLD="Y"+STR(I,4,0)
@2,20 SAY "YEAR: "+STR(I,4,0)
@6,3 SAY "DEMAND"
@7,3 SAY "-DOMESTIC"
@8,3 SAY "-INDUSTRIAL"
@10,3 SAY "TOTAL"
@12,3 SAY "SUPPLY"
@13,3 SAY "-NAT.FOREST"
@14,3 SAY "-PLANTATION"
@15,3 SAY "-VIL.FOREST"
@17,3 SAY "TOTAL"
@19,3 SAY "BALANCE"
@9,3 SAY HD2
@11,3 SAY HD2
@16,3 SAY HD2
@18,3 SAY HD2
@20,3 SAY HD2
```

```
FOR V=1 TO 8
    X=7
    IF V#8
        SEEK A_S1[V]
        @X,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_DOT[(I-Y_B)/Y_I+1]=
                    A_DOT[(I-Y_B)/Y_I+1]+&FLD
        A_D_T[V]=A_D_T[V]+&FLD
        SEEK A_S2[V]
        @X+1,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_INT[(I-Y_B)/Y_I+1]=
                    A_INT[(I-Y_B)/Y_I+1]+&FLD
        A_D_T[V]=A_D_T[V]+&FLD
        @X+3,6+8*V SAY A_D_T[V]
                              PICTURE "9999999"
        SEEK A_S3[V]
        @X+6,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_NFT[(I-Y_B)/Y_I+1]=
                    A_NFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        SEEK A_S4[V]
        @X+7,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_PLT[(I-Y_B)/Y_I+1]=
                    A_PLT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        SEEK A_S5[V]
        @X+8,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_VFT[(I-Y_B)/Y_I+1]=
                    A_VFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        @X+10,6+8*V SAY A_S_T[V]
                              PICTURE "9999999"
        @X+12,6+8*V SAY A_S_T[V]-
                      A_D_T[V] PICTURE "9999999"
    ELSE
        @X,6+8*V SAY A_DOT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
        @X+1,6+8*V SAY A_INT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
        A_D_T[V]=A_DOT[(I-Y_B)/Y_I+1]
                         +A_INT[(I-Y_B)/Y_I+1]
        @X+3,6+8*V SAY A_D_T[V]
                              PICTURE "99999999"
        @X+6,6+8*V SAY A_NFT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
        @X+7,6+8*V SAY A_PLT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
        @X+8,6+8*V SAY A_VFT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
        A_S_T[V]=A_NFT[(I-Y_B)/Y_I+1]
                       +A_PLT[(I-Y_B)/Y_I+1]+
                          A_VFT[(I-Y_B)/Y_I+1]
        @X+10,6+8*V SAY A_S_T[V]
                              PICTURE "99999999"
        @X+12,6+8*V SAY A_S_T[V]-
                      A_D_T[V] PICTURE "99999999"
    ENDIF
NEXT
X=19
I=I+Y_I
IF I<Y_B+Y_N+Y_I
    M_I=I
    P_CH=.F.
ELSE
    P_CH=.T.
ENDIF
```

```
RETURN
*!**************************************
*!      Procedure: BPGS2
*!      Calls
*!              : I_T8     (procedure in SIM.PRG)
*!              : I_TY     (procedure in SIM.PRG)
*!**************************************
PROC BPGS2
A_S1[1]="NW_SW"
A_S1[2]="NC_SW"
A_S1[3]="W_SW"
A_S1[4]="S_SW"
A_S1[5]="SE_SW"
A_S1[6]="NE_SW"
A_S1[7]="CHT_SW"
A_S2[1]="NW_PANEL"
A_S2[2]="NC_PANEL"
A_S2[3]="W_PANEL"
A_S2[4]="S_PANEL"
A_S2[5]="SE_PANEL"
A_S2[6]="NE_PANEL"
A_S2[7]="CHT_PANEL"
A_S3[1]="NF_T_NW"
A_S3[2]="NF_T_NC"
A_S3[3]="NF_T_W"
A_S3[4]="NF_T_S"
A_S3[5]="NF_T_SE"
A_S3[6]="NF_T_NE"
A_S3[7]="NF_T_CHT"
A_S4[1]="PL_T_NW"
A_S4[2]="PL_T_NC"
A_S4[3]="PL_T_W"
A_S4[4]="PL_T_S"
A_S4[5]="PL_T_SE"
A_S4[6]="PL_T_NE"
A_S4[7]="PL_T_CHT"
A_S5[1]="VF_T_NW"
A_S5[2]="VF_T_NC"
A_S5[3]="VF_T_W"
A_S5[4]="VF_T_S"
A_S5[5]="VF_T_SE"
A_S5[6]="VF_T_NE"
A_S5[7]="VF_T_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY "TIMBER"
@3,3 SAY "------"
@3,66 SAY "(in '000 m3)"
@4,3 SAY HD1
@5,3 SAY HD2
I=M_I&G
FLD="Y"+STR(I,4,0)
@2,20 SAY "YEAR: "+STR(I,4,0)
@6,3 SAY "DEMAND"
@7,3 SAY "-DOM.+COMM."
@8,3 SAY "-PANEL PR."
@10,3 SAY "TOTAL"
@12,3 SAY "SUPPLY"
@13,3 SAY "-NAT.FOREST"
@14,3 SAY "-PLANTATION"
@15,3 SAY "-VIL.FOREST"
@17,3 SAY "TOTAL"
@19,3 SAY "BALANCE"
@9,3 SAY HD2
@11,3 SAY HD2
@16,3 SAY HD2
```

```
@18,3 SAY HD2
@20,3 SAY HD2
FOR V=1 TO 8
    X=7
    IF V#8
        SEEK A_S1[V]
        @X,6+8*V SAY &FLD PICTURE
                                      "9999999"
        A_DOT[(I-Y_B)/Y_I+1]=
                    A_DOT[(I-Y_B)/Y_I+1]+&FLD
        A_D_T[V]=A_D_T[V]+&FLD
        SEEK A_S2[V]
        @X+1,6+8*V SAY &FLD PICTURE
                                      "9999999"
        A_INT[(I-Y_B)/Y_I+1]=
                    A_INT[(I-Y_B)/Y_I+1]+&FLD
        A_D_T[V]=A_D_T[V]+&FLD
        @X+3,6+8*V SAY A_D_T[V]
                           PICTURE "9999999"
        SEEK A_S3[V]
        @X+6,6+8*V SAY &FLD PICTURE
                                      "9999999"
        A_NFT[(I-Y_B)/Y_I+1]=
                    A_NFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        SEEK A_S4[V]
        @X+7,6+8*V SAY &FLD PICTURE
                                      "9999999"
        A_PLT[(I-Y_B)/Y_I+1]=
                    A_PLT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        SEEK A_S5[V]
        @X+8,6+8*V SAY &FLD PICTURE
                                      "9999999"
        A_VFT[(I-Y_B)/Y_I+1]=
                    A_VFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        @X+10,6+8*V SAY A_S_T[V]
                           PICTURE "9999999"
        @X+12,6+8*V SAY A_S_T[V]-
                    A_D_T[V] PICTURE "9999999"
    ELSE
        @X,6+8*V SAY A_DOT[(I-Y_B)/
                    Y_I+1] PICTURE "99999999"
        @X+1,6+8*V SAY A_INT[(I-Y_B)/
                    Y_I+1] PICTURE "99999999"
        A_D_T[V]=A_DOT[(I-Y_B)/
                    Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
        @X+3,6+8*V SAY A_D_T[V]
                           PICTURE "99999999"
        @X+6,6+8*V SAY A_NFT[(I-Y_B)/
                    Y_I+1] PICTURE "99999999"
        @X+7,6+8*V SAY A_PLT[(I-Y_B)/
                    Y_I+1] PICTURE "99999999"
        @X+8,6+8*V SAY A_VFT[(I-Y_B)/
                    Y_I+1] PICTURE "99999999"
        A_S_T[V]=A_NFT[(I-Y_B)/Y_I+1]
                    +A_PLT[(I-Y_B)/Y_I+1]+
                      A_VFT[(I-Y_B)/Y_I+1]
        @X+10,6+8*V SAY A_S_T[V]
                           PICTURE "99999999"
        @X+12,6+8*V SAY A_S_T[V]-
                    A_D_T[V] PICTURE "99999999"
    ENDIF
NEXT
X=19
I=I+Y_I
IF I<Y_B+Y_N+Y_I
    M_I=I
    P_CH=.F.
ELSE
    P_CH=.T.
ENDIF
RETURN
*!*************************************************
*!      Procedure: BPGS3
*!      Calls
*!            : I_T8     (procedure in SIM.PRG)
*!            : I_TY     (procedure in SIM.PRG)
*!*************************************************
PROC BPGS3
A_S1[1]="NW_PT_D"
A_S1[2]="NC_PT_D"
A_S1[3]="W_PT_D"
A_S1[4]="S_PT_D"
A_S1[5]="SE_PT_D"
A_S1[6]="NE_PT_D"
A_S1[7]="CHT_PT_D"
A_S2[1]="NW_PT_I"
A_S2[2]="NC_PT_I"
A_S2[3]="W_PT_I"
A_S2[4]="S_PT_I"
A_S2[5]="SE_PT_I"
A_S2[6]="NE_PT_I"
A_S2[7]="CHT_PT_I"
A_S3[1]="NF_P_NW"
A_S3[2]="NF_P_NC"
A_S3[3]="NF_P_W"
A_S3[4]="NF_P_S"
A_S3[5]="NF_P_SE"
A_S3[6]="NF_P_NE"
A_S3[7]="NF_P_CHT"
A_S4[1]="PL_P_NW"
A_S4[2]="PL_P_NC"
A_S4[3]="PL_P_W"
A_S4[4]="PL_P_S"
A_S4[5]="PL_P_SE"
A_S4[6]="PL_P_NE"
A_S4[7]="PL_P_CHT"
A_S5[1]="VF_P_NW"
A_S5[2]="VF_P_NC"
A_S5[3]="VF_P_W"
A_S5[4]="VF_P_S"
A_S5[5]="VF_P_SE"
A_S5[6]="VF_P_NE"
A_S5[7]="VF_P_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY  "POST & POLES"
@3,3 SAY  "------------"
@3,66 SAY "(in '000 m3)"
@4,3 SAY HD1
@5,3 SAY HD2
I=M_I&G
FLD="Y"+STR(I,4,0)
@2,20 SAY "YEAR: "+STR(I,4,0)
@6,3 SAY "DEMAND"
@7,3 SAY  "-DOMESTIC"
@8,3 SAY  "-INDUSTRIAL"
@10,3 SAY "TOTAL"
@12,3 SAY "SUPPLY"
@13,3 SAY "-NAT.FOREST"
@14,3 SAY "-PLANTATION"
@15,3 SAY "-VIL.FOREST"
@17,3 SAY "TOTAL"
@19,3 SAY "BALANCE"
@9,3 SAY HD2
```

```
@11,3 SAY HD2
@16,3 SAY HD2
@18,3 SAY HD2
@20,3 SAY HD2
FOR V=1 TO 8
    X=7
    IF V#8
        SEEK A_S1[V]
        @X,6+8*V SAY &FLD
                                PICTURE "9999999"
        A_DOT[(I-Y_B)/Y_I+1]=
                    A_DOT[(I-Y_B)/Y_I+1]+&FLD
        A_D_T[V]=A_D_T[V]+&FLD
        SEEK A_S2[V]
        @X+1,6+8*V SAY &FLD
                                PICTURE "9999999"
        A_INT[(I-Y_B)/Y_I+1]=
                    A_INT[(I-Y_B)/Y_I+1]+&FLD
        A_D_T[V]=A_D_T[V]+&FLD
        @X+3,6+8*V SAY A_D_T[V]
                                PICTURE "9999999"
        SEEK A_S3[V]
        @X+6,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_NFT[(I-Y_B)/Y_I+1]=
                    A_NFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        SEEK A_S4[V]
        @X+7,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_PLT[(I-Y_B)/Y_I+1]=
                    A_PLT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        SEEK A_S5[V]
        @X+8,6+8*V SAY &FLD PICTURE
                                        "9999999"
        A_VFT[(I-Y_B)/Y_I+1]=
                    A_VFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
        @X+10,6+8*V SAY A_S_T[V]
                                PICTURE "9999999"
        @X+12,6+8*V SAY A_S_T[V]-
                    A_D_T[V] PICTURE "9999999"
  - ELSE
        @X,6+8*V SAY A_DOT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
        @X+1,6+8*V SAY A_INT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
        A_D_T[V]=A_DOT[(I-Y_B)
                 /Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
        @X+3,6+8*V SAY A_D_T[V]
                                PICTURE "99999999"
        @X+6,6+8*V SAY A_NFT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
        @X+7,6+8*V SAY A_PLT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
        @X+8,6+8*V SAY A_VFT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
        A_S_T[V]=A_NFT[(I-Y_B)/Y_I+1]
                         +A_PLT[(I-Y_B)/Y_I+1]+
                           A_VFT[(I-Y_B)/Y_I+1]
        @X+10,6+8*V SAY A_S_T[V]
                                PICTURE "99999999"
        @X+12,6+8*V SAY A_S_T[V]-
                   A_D_T[V] PICTURE "99999999"
    ENDIF
NEXT
X=19
I=I+Y_I
IF I<Y_B+Y_N+Y_I
    M_I=I
```

```
    P_CH=.F.
ELSE
    P_CH=.T.
ENDIF
RETURN
*!**************************************
*!    Procedure: BPGS4
*!    Calls
*!    : I_T8    (procedure in SIM.PRG)
*!    : I_TY    (procedure in SIM.PRG)
*!**************************************
PROC BPGS4
A_S1[1]="NW_NP"
A_S1[2]="NC_NP"
A_S1[3]="W_NP"
A_S1[4]="S_NP"
A_S1[5]="SE_NP"
A_S1[6]="NE_NP"
A_S1[7]="CHT_NP"
A_S2[1]="NW_PPR"
A_S2[2]="NC_PPR"
A_S2[3]="W_PPR"
A_S2[4]="S_PPR"
A_S2[5]="SE_PPR"
A_S2[6]="NE_PPR"
A_S2[7]="CHT_PPR"
A_S3[1]="NF_PW_NW"
A_S3[2]="NF_PW_NC"
A_S3[3]="NF_PW_W"
A_S3[4]="NF_PW_S"
A_S3[5]="NF_PW_SE"
A_S3[6]="NF_PW_NE"
A_S3[7]="NF_PW_CHT"
A_S4[1]="PL_PW_NW"
A_S4[2]="PL_PW_NC"
A_S4[3]="PL_PW_W"
A_S4[4]="PL_PW_S"
A_S4[5]="PL_PW_SE"
A_S4[6]="PL_PW_NE"
A_S4[7]="PL_PW_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY  "PULPWOOD - POPULATION
          GROWTH WITH CONSTANT LITERACY"
@3,3 SAY  "--------------------
                        ----------------------------"
@3,66 SAY "(in '000 m3)"
@4,3 SAY HD1
@5,3 SAY HD2
I=M_I&G
FLD="Y"+STR(I,4,0)
@2,66 SAY "YEAR: "+STR(I,4,0)
@6,3 SAY "DEMAND"
@7,3 SAY  "-NEWSPRINT"
@8,3 SAY  "-OTHERS"
@10,3 SAY "TOTAL"
@12,3 SAY "SUPPLY"
@13,3 SAY "-NAT.FOREST"
@14,3 SAY "-PLANTATION"
@16,3 SAY "TOTAL"
@18,3 SAY "BALANCE"
@9,3 SAY HD2
@11,3 SAY HD2
@15,3 SAY HD2
@17,3 SAY HD2
@19,3 SAY HD2
```

```
FOR V=1 TO 8
     X=7
     IF V#8
          IF V=1
               SEEK A_S1[V]
                A_DOT[(I-Y_B)/Y_I+1]=
              A_DOT[(I-Y_B)/Y_I+1]+&FLD*50/35.3
               A_D_T[V]=A_D_T[V]
                                        +&FLD*50/35.3

               SEEK A_S2[V]
               A_INT[(I-Y_B)/Y_I+1]=
               A_INT[(I-Y_B)/Y_I+1]+&FLD*50/35.3
               A_D_T[V]=A_D_T[V]+
                                        &FLD*50/35.3
          ENDIF
          SEEK A_S3[V]
          @X+6,6+8*V SAY &FLD PICTURE
                                             "9999999"
          A_NFT[(I-Y_B)/Y_I+1]=
                         A_NFT[(I-Y_B)/Y_I+1]+&FLD
          A_S_T[V]=A_S_T[V]+&FLD
          SEEK A_S4[V]
          @X+7,6+8*V SAY &FLD PICTURE
                                             "9999999"
          A_PLT[(I-Y_B)/Y_I+1]=
                         A_PLT[(I-Y_B)/Y_I+1]+&FLD
          A_S_T[V]=A_S_T[V]+&FLD
          @X+9,6+8*V SAY A_S_T[V]
                                   PICTURE "9999999"
     ELSE
          @X,6+8*V SAY A_DOT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
          @X+1,6+8*V SAY A_INT[(I-Y_B)
                         /Y_I+1] PICTURE "99999999"
          A_D_T[V]=A_DOT[(I-Y_B)
                    /Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
          @X+3,6+8*V SAY A_D_T[V]
                                   PICTURE "99999999"
          @X+6,6+8*V SAY A_NFT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
          @X+7,6+8*V SAY A_PLT[(I-Y_B)/
                         Y_I+1] PICTURE "99999999"
          A_S_T[V]=A_NFT[(I-Y_B)/
                    Y_I+1]+A_PLT[(I-Y_B)/Y_I+1]
          @X+9,6+8*V SAY A_S_T[V]
                                   PICTURE "99999999"
          @X+11,6+8*V SAY A_S_T[V]-
                    A_D_T[V] PICTURE "99999999"
     ENDIF
NEXT
X=19
I=I+Y_I
IF I<Y_B+Y_N+Y_I
     M_I=I
     P_CH=.F.
ELSE
     P_CH=.T.
ENDIF
RETURN
*!***************************************
*!   Procedure: BPGS5
*!   Calls
*!        : I_T8     (procedure in SIM.PRG)
*!        : I_TY     (procedure in SIM.PRG)
*!***************************************
PROC BPGS5
A_S1[1]="NW_NP"
A_S1[2]="NC_NP"
A_S1[3]="W_NP"
A_S1[4]="S_NP"
A_S1[5]="SE_NP"
```

```
A_S1[6]="NE_NP"
A_S1[7]="CHT_NP"
A_S2[1]="NW_PPR"
A_S2[2]="NC_PPR"
A_S2[3]="W_PPR"
A_S2[4]="S_PPR"
A_S2[5]="SE_PPR"
A_S2[6]="NE_PPR"
A_S2[7]="CHT_PPR"
A_S3[1]="NF_PW_NW"
A_S3[2]="NF_PW_NC"
A_S3[3]="NF_PW_W"
A_S3[4]="NF_PW_S"
A_S3[5]="NF_PW_SE"
A_S3[6]="NF_PW_NE"
A_S3[7]="NF_PW_CHT"
A_S4[1]="PL_PW_NW"
A_S4[2]="PL_PW_NC"
A_S4[3]="PL_PW_W"
A_S4[4]="PL_PW_S"
A_S4[5]="PL_PW_SE"
A_S4[6]="PL_PW_NE"
A_S4[7]="PL_PW_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY "PULPWOOD - POPULATION
                              AND LITERACY GROWTH"
@3,3 SAY "--------------------------
                                    --------------"
@3,66 SAY "(in '000 m3)"
@4,3 SAY HD1
@5,3 SAY HD2
I=M_I&G
FLD="Y"+STR(I,4,0)
@2,66 SAY "YEAR: "+STR(I,4,0)
@6,3 SAY "DEMAND"
@7,3 SAY "-NEWSPRINT"
@8,3 SAY "-OTHERS"
@10,3 SAY "TOTAL"
@12,3 SAY "SUPPLY"
@13,3 SAY "-NAT.FOREST"
@14,3 SAY "-PLANTATION"
@16,3 SAY "TOTAL"
@18,3 SAY "BALANCE"
@9,3 SAY HD2
@11,3 SAY HD2
@15,3 SAY HD2
@17,3 SAY HD2
@19,3 SAY HD2
FOR V=1 TO 8
     X=7
     IF V#8
          IF V=2
               SEEK A_S1[V]
               A_DOT[(I-Y_B)/Y_I+1]=
             A_DOT[(I-Y_B)/Y_I+1]+&FLD*50/35.3
               A_D_T[V]=A_D_T[V]+&FLD*
                                          50/35.3
               SEEK A_S2[V]
               A_INT[(I-Y_B)/Y_I+1]=
              A_INT[(I-Y_B)/Y_I+1]+&FLD*50/35.3
               A_D_T[V]=A_D_T[V]+&FLD*
                                          50/35.3
          ENDIF
          SEEK A_S3[V]
          @X+6,6+8*V SAY &FLD PICTURE
```

```
                                        "9999999"
            A_NFT[(I-Y_B)/Y_I+1]=
                        A_NFT[(I-Y_B)/Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
            SEEK A_S4[V]
            @X+7,6+8*V SAY &FLD PICTURE
                                        "9999999"
            A_PLT[(I-Y_B)/Y_I+1]=
                        A_PLT[(I-Y_B)/Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
            @X+9,6+8*V SAY A_S_T[V]
                              PICTURE "9999999"
      ELSE
            @X,6+8*V SAY A_DOT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
            @X+1,6+8*V SAY A_INT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
            A_D_T[V]=A_DOT[(I-Y_B)/
                        Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
            @X+3,6+8*V SAY A_D_T[V]
                              PICTURE "99999999"
            @X+6,6+8*V SAY A_NFT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
            @X+7,6+8*V SAY A_PLT[(I-Y_B)/
                        Y_I+1] PICTURE "99999999"
            A_S_T[V]=A_NFT[(I-Y_B)/
                        Y_I+1]+A_PLT[(I-Y_B)/Y_I+1]
            @X+9,6+8*V SAY A_S_T[V]
                              PICTURE "99999999"
            @X+11,6+8*V SAY A_S_T[V]-
                        A_D_T[V] PICTURE "99999999"
      ENDIF
NEXT
X=19
I=I+Y_I
IF I<Y_B+Y_N+Y_I
      M_I=I
      P_CH=.F.
ELSE
      P_CH=.T.
ENDIF
RETURN
*!**************************************
*!    Procedure: D_PR
*!    Called by: DEMAND         (proc in SIM.PRG)
*!    Calls
*!    : D_PR1  (procedure in SIM.PRG)
*!    : D_PR2  (procedure in SIM.PRG)
*!**************************************
PROC D_PR
DO WHILE P2#0
      @10,31 TO 13,52 DOUBLE
      @11,32 PROMPT " INITIAL
                                  PARAMETERS "

      @12,32 PROMPT " SIMULATION
                                     RESULTS "
      MENU TO P2
      IF P2#0
            SAVE SCREEN TO SCR3
            DO CASE
            CASE P2=1
                  DO D_PR1
            CASE P2=2
                  DO D_PR2
            ENDCASE
            CLOSE DATABASES
            REST SCREEN FROM SCR3
      ENDIF
ENDDO
```

```
RETURN
*!**************************************
*!    Procedure: D_PR1
*!    Called by: D_PR          (procedure in SIM.PRG)
*!    Calls
*!          : INI_1   (procedure in SIM.PRG)
*!          : INI_3   (procedure in SIM.PRG)
*!          : INI_2   (procedure in SIM.PRG)
*!    Uses
*!          : &F_NAME2
*!          : &F_NAME1
*!    Indexes
*!          : &F_NAME2
*!          : &F_NAME1
*!**************************************
PROC D_PR1
SELE A
USE &F_NAME2 INDEX &F_NAME2
SELE B
USE &F_NAME1 INDEX &F_NAME1
SELE B
DECLARE A_FWD1[5]
DECLARE A_FWD2[5]
DECLARE A_PT1[5]
DECLARE A_PT2[5]
DECLARE A_ST[7]
DECLARE A_NW[5]
DECLARE A_PA[5]
DECLARE A_PNL[7]
HD= "INITIAL PARAMETERS FOR DEMAND
SIMULATION"
SEEK "FULWD_DU"
DO INI_1 WITH A_FWD1
SEEK "FULWD_IU"
DO INI_1 WITH A_FWD2
SEEK "PT/PL_DU"
DO INI_1 WITH A_PT1
SEEK "PT/PL_IU"
DO INI_1 WITH A_PT2
SEEK "STMBR"
DO INI_3 WITH A_ST
SEEK "NWPT"
DO INI_1 WITH A_NW
SEEK "PAPR"
DO INI_1 WITH A_PA
SEEK "PANEL"
DO INI_3 WITH A_PNL
SET DEVICE TO PRINT
@ 2,3 SAY HD
@ 4,2 SAY "[ CONSUMPTION IN THE
                        YEAR "+STR(Y_B,4,0)+" ]"
@ 6,2 SAY "               ------PER
                  CAPITA CONSUMPTION------|
              GOVT.& COMM.|CONSUMPTION"
@ 7,2 SAY "                   |
                 --- URBAN----|----RURAL----|
              CONSUMPTION |GROWTH RATE"
@ 8,2 SAY "               GROSS |
                RICH | POOR | RICH | POOR |
                     (% of urban)|(% per yr)"
@ 9,2 SAY "               ------|------|------|
                 ------|------|------------|-----------"
@10,2 SAY "FUELWOOD"
@12,2 SAY "->DOMESTIC (m3):"
FOR X=2 TO 5
      @12,14+X*7 SAY A_FWD1[X] PICTURE
                                   "9.9999"
NEXT
@12,71 SAY A_FWD1[1] PICTURE "999.999"
@14,2 SAY "->INDUSTRIAL(m3): "+
                          STR(A_FWD2[2],6,4)
```

```
@14,71 SAY A_FWD2[1] PICTURE "999.999"
@16,2 SAY  "SAWN TIMBER  (m3):"
FOR X=1 TO 2
     @16,21+X*7 SAY A_ST[X] PICTURE
                                              "9.9999"
NEXT
@16,25+3*7 SAY A_ST[3] PICTURE "9.9999"
@16,59 SAY A_ST[4] PICTURE "999.999"
@16,71 SAY A_ST[5] PICTURE "999.999"
@18,2 SAY  "->RWE FACTOR (%):"
@18,21 SAY A_ST[6] PICTURE "99.999"
@20,2 SAY "POSTS & POLES"
@22,2 SAY "->DOMESTIC  (m3): "+
                                    STR(A_PT1[2],6,4)
@22,71 SAY A_PT1[1] PICTURE "999.999"
@24,2 SAY "->INDUSTRIAL(m3): "
                                   +STR(A_PT2[2],6,4)
@24,71 SAY A_PT2[1] PICTURE "999.999"
@26,2 SAY "NEWSPRINT    (kg): "
                                   +STR(A_NW[2],6,4)
@26,71 SAY A_NW[1] PICTURE "999.999"

@28,2 SAY "OTHER PAPERS (kg): "+
                                    STR(A_PA[2],6,4)
@28,71 SAY A_PA[1] PICTURE "999.999"
@30,2 SAY "             NORTH
          NORTH              SOUTH  NORTH"
@31,2 SAY "                  WEST
                    CENTRAL  WEST  SOUTH
                             EAST  EAST   CHT"
@32,2 SAY "                  ------- ------- -------
                             ------- ------- ------- -------"
@34,2 SAY "PANEL PRODUCTS(m3):"
FOR X=1 TO 7
     @34,14+X*8 SAY A_PNL[X] PICTURE
                                            "999.999"
NEXT
DECLARE A_P_T[Y_R+1]
DECLARE A_PU1[Y_R+1]
DECLARE A_PU2[Y_R+1]
DECLARE A_PU3[Y_R+1]
DECLARE A_PU4[Y_R+1]
DECLARE A_PU5[Y_R+1]
DECLARE A_PU6[Y_R+1]
DECLARE A_PU7[Y_R+1]
DECLARE A_LIT[Y_R+1]
DECLARE A_P11[Y_R+1]
DECLARE A_P12[Y_R+1]
DECLARE A_P13[Y_R+1]
DECLARE A_P14[Y_R+1]
DECLARE A_P21[Y_R+1]
DECLARE A_P22[Y_R+1]
DECLARE A_P23[Y_R+1]
DECLARE A_P24[Y_R+1]
DECLARE A_P31[Y_R+1]
DECLARE A_P32[Y_R+1]
DECLARE A_P33[Y_R+1]
DECLARE A_P34[Y_R+1]
DECLARE A_P41[Y_R+1]
DECLARE A_P42[Y_R+1]
DECLARE A_P43[Y_R+1]
DECLARE A_P44[Y_R+1]
DECLARE A_P51[Y_R+1]
DECLARE A_P52[Y_R+1]
DECLARE A_P53[Y_R+1]
DECLARE A_P54[Y_R+1]
DECLARE A_P61[Y_R+1]
DECLARE A_P62[Y_R+1]
DECLARE A_P63[Y_R+1]
DECLARE A_P64[Y_R+1]
DECLARE A_P71[Y_R+1]
```

```
DECLARE A_P72[Y_R+1]
DECLARE A_P73[Y_R+1]
DECLARE A_P74[Y_R+1]
DECLARE A_LIT[Y_R+1]
SELE A
SEEK "T_POP"
DO INI_2 WITH A_P_T
SEEK "U_POP_1"
DO INI_2 WITH A_PU1
SEEK "U_POP_2"
DO INI_2 WITH A_PU2
SEEK "U_POP_3"
DO INI_2 WITH A_PU3
SEEK "U_POP_4"
DO INI_2 WITH A_PU4
SEEK "U_POP_5"
DO INI_2 WITH A_PU5
SEEK "U_POP_6"
DO INI_2 WITH A_PU6
SEEK "U_POP_7"
DO INI_2 WITH A_PU7
SEEK "LITERACY"
DO INI_2 WITH A_LIT
SEEK "NW_POP_1"
DO INI_2 WITH A_P11
SEEK "NW_POP_2"
DO INI_2 WITH A_P12
SEEK "NW_POP_3"
DO INI_2 WITH A_P13
SEEK "NW_POP_4"
DO INI_2 WITH A_P14
SEEK "NC_POP_1"
DO INI_2 WITH A_P21
SEEK "NC_POP_2"
DO INI_2 WITH A_P22
SEEK "NC_POP_3"
DO INI_2 WITH A_P23
SEEK "NC_POP_4"
DO INI_2 WITH A_P24
SEEK "W_POP_1"
DO INI_2 WITH A_P31
SEEK "W_POP_2"
DO INI_2 WITH A_P32
SEEK "W_POP_3"
DO INI_2 WITH A_P33
SEEK "W_POP_4"
DO INI_2 WITH A_P34
SEEK "S_POP_1"
DO INI_2 WITH A_P41
SEEK "S_POP_2"
DO INI_2 WITH A_P42
SEEK "S_POP_3"
DO INI_2 WITH A_P43
SEEK "S_POP_4"
DO INI_2 WITH A_P44
SEEK "SE_POP_1"
DO INI_2 WITH A_P51
SEEK "SE_POP_2"
DO INI_2 WITH A_P52
SEEK "SE_POP_3"
DO INI_2 WITH A_P53
SEEK "SE_POP_4"
DO INI_2 WITH A_P54
SEEK "NE_POP_1"
DO INI_2 WITH A_P61
SEEK "NE_POP_2"
DO INI_2 WITH A_P62
SEEK "NE_POP_3"
DO INI_2 WITH A_P63
SEEK "NE_POP_4"
DO INI_2 WITH A_P64
```

```
SEEK "CHT_POP_1"
DO INI_2 WITH A_P71
SEEK "CHT_POP_2"
DO INI_2 WITH A_P72
SEEK "CHT_POP_3"
DO INI_2 WITH A_P73
SEEK "CHT_POP_4"
DO INI_2 WITH A_P74
HD1="[ POPULATION ]"
HD2="YEAR  POPULATION LITERACY
                          ---- URBAN POPULATION
                               (% of regional total) ----"
HD3="    (in '000s)  (%)     N.WEST
                          N.CEN.  WEST  SOUTH
                            S.EAST N.EAST  CHT"
HD4="---- --------- --------  ------ ------ ------
                              ------ ------ ------ ------"
@ 38,3 SAY HD1
@ 40,3 SAY HD2
@ 41,3 SAY HD3
@ 42,3 SAY HD4
X=43
V=2
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    @X,3 SAY STR(I,4,0)
    @X,10 SAY A_P_T[V] PICTURE
                                          "9999999"
    @X,20 SAY A_LIT[V] PICTURE
                                          "999.999"
    FOR YY=1 TO 7
        Y=STR(YY,1,0)
        @X,23+YY*7 SAY A_PU&Y[V]
                               PICTURE "999.99"
    NEXT
    X=X+2
    V=V+1
    IF X=57.AND.I<Y_B+Y_N+Y_I
        @ 2,3 SAY HD
        @ 4,3 SAY HD1
        @ 6,3 SAY HD2
        @ 7,3 SAY HD3
        @ 8,3 SAY HD4
        X=9
    ENDIF
NEXT
@ X+1,3 SAY " ----------- POPULATION
                   DISTRIBUTION (as % of total
                                  population) ------"
@ X+2,3 SAY "  RICH    N.WEST
                   N.CENT.  WEST    SOUTH
                      S.EAST  N.EAST    CHT"
@ X+3,3 SAY " ------- ------- ------- -------
                      ------- ------- ------- -------"
@ X+4,5 SAY A_P_T[1] PICTURE "999.999"
P='1'
FOR YY=1 TO 7
    Y=STR(YY,1,0)
    @X+4,5+YY*9 SAY A_P&Y&P[1]
                                PICTURE "999.999"
NEXT
HD1="[ POPULATION ('000s) ]"
HD2="                NORTH   NORTH
                               SOUTH   NORTH"
HD3="YEAR              WEST  CENTRAL
         WEST  SOUTH   EAST   EAST     CHT"
HD4="----            ------- ------- ------- -------
                               ------- ------- -------"
@ 2,3 SAY HD
@ 4,3 SAY HD1
@ 6,3 SAY HD2
@ 7,3 SAY HD3
```

```
@ 8,3 SAY HD4
XX=8
V=2
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FOR Y=1 TO 4
        FOR X=1 TO 7
            P=STR(X,1,0)
            IF X=1
                DO CASE
                CASE Y=1
                    @XX+2,3 SAY STR(I,4,0)+
                                 "  URBAN-RICH:"
                CASE Y=2
                    @XX+4,3 SAY
                           "               -POOR:"
                CASE Y=3
                    @XX+6,3 SAY
                           "          RURAL-RICH:"
                CASE Y=4
                    @XX+8,3 SAY
                           "               -POOR:"
                ENDCASE
            ENDIF
            YY=STR(Y,1,0)

            @XX+Y*2,14+X*8 SAY
              A_P&P&YY[V] PICTURE "9999999"
        NEXT
    NEXT
    @XX+10,3 SAY "            -------
               ------- ------- ------- -------
                                ------- -------"
    V=V+1
    XX=XX+10
    IF XX=58.AND.I<Y_B+Y_N
        @ 2,3 SAY HD
        @ 4,3 SAY HD1
        @ 6,3 SAY HD2
        @ 7,3 SAY HD3
        @ 8,3 SAY HD4
        XX=8
    ENDIF
NEXT
@64,3 SAY "Date: "+DTOC(DATE())
@0,0 SAY " "
SET DEVICE TO SCREEN
RETURN
*!****************************************
*!    Procedure: D_PR2X
*!    Called by: D_PR2          (proc in SIM.PRG)
*!    Calls: I_TOT         (procedure in SIM.PRG)
*!****************************************
PROC D_PR2X
DO I_TOT
IF X>45
    X=0
    @X+2,3 SAY HD
ENDIF
@X+4,3 SAY HD3
@X+5,3 SAY HD4
@X+5,66 SAY "(in '000 m3)"
@X+7,3 SAY HD1
@X+8,3 SAY HD2
X=X+9
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FLD="Y"+STR(I,4,0)
    FOR V=1 TO 8
        IF V#8
            SEEK A_S1[V]
            IF V=1
                @X,3 SAY
```

```
                                        "YEAR-"+STR(I,4,0)
                ENDIF
                @X,6+8*V SAY &FLD
                                        PICTURE "9999.99"
                TOT[(I-Y_B)/Y_I+1]=
                            TOT[(I-Y_B)/Y_I+1]+&FLD
            ELSE

                @X,6+8*V SAY TOT[(I-Y_B)/
                            Y_I+1] PICTURE "99999.99"
            ENDIF
        NEXT
        X=X+2
        IF X=57.AND.I<Y_B+Y_N
            @2,3 SAY HD
            @4,3 SAY HD3
            @5,3 SAY HD4
            @5,66 SAY "(in '000 m3)"
            @7,3 SAY HD1
            @8,3 SAY HD2
            X=9
        ENDIF
    NEXT
    RETURN
    *!**************************************
    *!    Procedure: D_PR2Y
    *!    Called by: D_PR2          (procedure in SIM.PRG)
    *!    Calls: I_TOT              (procedure in SIM.PRG)
    *!**************************************
    PROC D_PR2Y
    DO I_TOT
    IF X>45
        X=0
        @X+2,3 SAY HD
    ENDIF
    @X+4,3 SAY  HD3
    @X+5,3 SAY  HD4
    @X+7,20 SAY HD5
    @X+8,20 SAY HD6
    @X+9,20 SAY HD7
    @X+10,20 SAY HD8
    X=X+11
    FOR I=Y_B TO Y_B+Y_N STEP Y_I
        FLD="Y"+STR(I,4,0)
        FOR V=1 TO 2
            SEEK A_S1[V]
            IF V=1
                @X,3 SAY "YEAR-"+STR(I,4,0)
            ENDIF
            @X,22+28*(V-1) SAY &FLD/1000
                                    PICTURE "9999.99"
            @X,34+28*(V-1) SAY &FLD/1000*
                            50/35.3 PICTURE "9999.99"
        NEXT
        X=X+2
        IF X=57.AND.I<Y_B+Y_N
            @2,3 SAY HD
            @4,3 SAY  HD3
            @5,3 SAY  HD4
            @7,20 SAY HD5
            @8,20 SAY HD6
            @9,20 SAY HD7
            @10,20 SAY HD8
            X=11
        ENDIF
    NEXT
    RETURN
    *!**************************************
    *!    Procedure: D_PR2
    *!    Called by: D_PR           (procedure in SIM.PRG)
    *!    Calls
```

```
              : D_PR2X      (procedure in SIM.PRG)
*!            : D_PR2Y      (procedure in SIM.PRG)
*!      Uses: &F_NAME2
*!      Indexes: &F_NAME2
*!***************************************
PROC D_PR2
SELE A
USE &F_NAME2 INDEX &F_NAME2
HD="FUTURE  PROJECTIONS  FROM  DEMAND
SIMULATION"
HD1="          N.WEST N.CENTR.  WEST  SOUTH
S.EAST  N.EAST   CHT   NATIONAL"
HD2="          ------- ------- ------- ------- -------
                              ------- ------- -------"

DECLARE A_S1[7]
DECLARE TOT[Y_N+1]
A_S1[1]="NW_FWD_D"
A_S1[2]="NC_FWD_D"
A_S1[3]="W_FWD_D"
A_S1[4]="S_FWD_D"
A_S1[5]="SE_FWD_D"
A_S1[6]="NE_FWD_D"
A_S1[7]="CHT_FWD_D"
SET DEVICE TO PRINT
HD3="FUELWOOD - DOMESTIC USE"
HD4="-----------------------"
X=50
DO D_PR2X
A_S1[1]="NW_FWD_I"
A_S1[2]="NC_FWD_I"
A_S1[3]="W_FWD_I"
A_S1[4]="S_FWD_I"
A_S1[5]="SE_FWD_I"
A_S1[6]="NE_FWD_I"
A_S1[7]="CHT_FWD_I"
HD3="FUELWOOD - INDUSTRIAL USE"
HD4="-------------------------"
DO D_PR2X
A_S1[1]="NW_SW"
A_S1[2]="NC_SW"
A_S1[3]="W_SW"
A_S1[4]="S_SW"
A_S1[5]="SE_SW"
A_S1[6]="NE_SW"
A_S1[7]="CHT_SW"
HD3="SAWN TIMBER (RWE)"
HD4="-----------------"
DO D_PR2X
A_S1[1]="NW_PT_D"
A_S1[2]="NC_PT_D"
A_S1[3]="W_PT_D"
A_S1[4]="S_PT_D"
A_S1[5]="SE_PT_D"
A_S1[6]="NE_PT_D"
A_S1[7]="CHT_PT_D"
HD3="POST & POLES - DOMESTIC USE"
HD4="---------------------------"
DO D_PR2X
A_S1[1]="NW_PT_I"
A_S1[2]="NC_PT_I"
A_S1[3]="W_PT_I"
A_S1[4]="S_PT_I"
A_S1[5]="SE_PT_I"
A_S1[6]="NE_PT_I"
A_S1[7]="CHT_PT_I"
HD3="POST & POLES - INDUSTRIAL USE"
HD4="-----------------------------"
DO D_PR2X
A_S1[1]="NW_PANEL"
A_S1[2]="NC_PANEL"
A_S1[3]="W_PANEL"
```

```
A_S1[4]="S_PANEL"
A_S1[5]="SE_PANEL"
A_S1[6]="NE_PANEL"
A_S1[7]="CHT_PANEL"
HD3="PANEL PRODUCTS"
HD4="--------------"
DO D_PR2X
A_S1[1]="NW_NP"
A_S1[2]="NC_NP"
A_S1[3]="W_NP"
A_S1[4]="S_NP"
A_S1[5]="SE_NP"
A_S1[6]="NE_NP"
A_S1[7]="CHT_NP"
HD3="NEWSPRINT PAPER"
HD4="---------------"
HD5="CONSIDERING    POPULATION
CONSIDERING POPULATION"
HD6="GROWTH (Fixed literacy)      AND
                              LITERACY GROWTH"
HD7="('000 M.Ton)  ('000 m3)     ('000
                                 M.Ton)  ('000 m3)"
HD8="------------ -----------    ------------ -----------"
DO D_PR2Y
A_S1[1]="NW_PPR"
A_S1[2]="NC_PPR"
A_S1[3]="W_PPR"
A_S1[4]="S_PPR"
A_S1[5]="SE_PPR"
A_S1[6]="NE_PPR"
A_S1[7]="CHT_PPR"
HD3="OTHER PAPERS"
HD4="------------"
DO D_PR2Y
@64,3 SAY "Date: "+DTOC(DATE())
@0,0 SAY " "
SET DEVICE TO SCREEN
RETURN
*!**************************************
*!    Procedure: S_PR
*!    Called by: SUPPLY            (proc in SIM.PRG)
*!    Calls
*!         : S_PR1   (procedure in SIM.PRG)
*!         : S_PR2   (procedure in SIM.PRG)
*!**************************************
PROC S_PR
DO WHILE P2# 0
    @10,47 TO 13,68 DOUBLE
    @11,48 PROMPT " INITIAL
                                  PARAMETERS "
    @12,48 PROMPT " SIMULATION
                                      RESULTS "
    MENU TO P2
    IF P2# 0
        SAVE SCREEN TO SCR3
        DO CASE
        CASE P2=1
            DO S_PR1
        CASE P2=2
            DO S_PR2
        ENDCASE
        CLOSE DATABASES
        REST SCREEN FROM SCR3
    ENDIF
ENDDO
RETURN
*!**************************************
*!    Procedure: S_PR1
*!    Called by: S_PR          (procedure in SIM.PRG)
*!    Calls
*!         : INI_3   (procedure in SIM.PRG)
*!         : INI_5   (procedure in SIM.PRG)
*!    Uses: &F_NAME1
*!    Indexes: &F_NAME1
*!**************************************
PROC S_PR1
SELE A
USE &F_NAME1 INDEX &F_NAME1
DECLARE A_NF1[12]
DECLARE A_NF2[12]
DECLARE A_NF3[12]
DECLARE A_NF4[12]
DECLARE A_NF5[12]
DECLARE A_NF6[12]
DECLARE A_PL1[12]
DECLARE A_PL2[12]
DECLARE A_PL3[12]
DECLARE A_PL4[12]
DECLARE A_PL5[12]
DECLARE A_PL6[12]
DECLARE A_V1[7]
DECLARE A_V2[7]
DECLARE A_V3[7]
DECLARE A_V4[7]
DECLARE A_V5[7]
DECLARE A_V6[7]
DECLARE A_RC[6]
A_RC[1]="VF_TMBR"
A_RC[2]="VF_POLE"
A_RC[3]="VF_FUEL"
A_RC[4]="VF_GR_T"
A_RC[5]="VF_GR_P"
A_RC[6]="VF_GR_F"
USE &F_NAME1 INDEX &F_NAME1
FOR X=1 TO 6
    XX=LTRIM(STR(X,2,0))
    SEEK A_RC[X]
    DO INI_3 WITH A_V&XX
NEXT
HD='INITIAL PARAMETERS FOR SUPPLY
                                     SIMULATION'
SET DEVICE TO PRINT
@2,3 SAY HD
L=0
HD1=" NATURAL FOREST >"
HDS="   NF "
FOR  NF=1 TO  7
    FOR R=1 TO Y_NF/6
        FOR XX=1 TO 6
            X=STR(XX,1,0)
            Y=LTRIM(STR((R-1)*
                                6+XX,2,0))
            DO INI_5 WITH "NF"+Y,
                          Z_STR[NF],A_NF&X
            HD1=HD1+HDS+STR((R-1)*
                                6+XX,2,0)
        NEXT
        @L+5,3 SAY "[ REGION: "+
                RTRIM(SUBSTR(C[NF],6))+",
                  YEAR: "+STR(Y_B,4,0)+" ]"
        @L+7,3 SAY HD1
        @L+8,3 SAY "------------------  -------
                ------- ------- ------- ------- -------"
        @L+9,3 SAY  "Productive area(ha):"
        FOR XX=1 TO 6
            X=STR(XX,1,0)
            @L+9,15+9*XX SAY
                  A_NF&X[1] PICTURE "9999999"
        NEXT
        FOR Z=2 TO 12
            DO CASE
            CASE Z=2
            @L+10,3 SAY
                          "MAI      (m3/ha/yr):"
        CASE Z=3
            @L+11,3 SAY
                          "Age            (yr):"
        CASE Z=4
            @L+12,3 SAY
                          "Crop density       :"
        CASE Z=5
            @L+13,3 SAY
                          "Working Cycle. (yr):"
        CASE Z=6
            @L+15,3 SAY
                          "Yield rate  (m3/ha):"
        CASE Z=7
            @L+17,3 SAY
                          "Yield distribution"
            @L+18,3 SAY
                          "   - Timber   (%):"
        CASE Z=8
            @L+19,3 SAY
                          "   - Pole     (%):"
        CASE Z=9
            @L+20,3 SAY
                          "   - Fuel     (%):"
        CASE Z=10
            @L+21,3 SAY
                          "   - Pulp    -(%):"
        CASE Z=11
            @L+23,3 SAY
                                   "Increase rate"
            @L+24,3 SAY
                          " -Prod. area (%/yr):"
        CASE Z=12
            @L+25,3 SAY
                          " -Yield      (%/yr):"
        ENDCASE
        FOR XX=1 TO 6
            X=STR(XX,1,0)
            @PROW(),15+9*XX SAY
              A_NF&X[Z] PICTURE "999.999"
        NEXT
    NEXT
    IF L=0
        L=28
    ELSE
        L=0
        @2,3 SAY HD
    ENDIF
  NEXT
NEXT
@2,3 SAY HD
L=0
HD1="  PLANTATION   >"
HDS="  PL "
FOR  PL=1 TO  7
    FOR R=1 TO Y_PL/6
        FOR XX=1 TO 6
            X=STR(XX,1,0)
            Y=LTRIM(STR((R-1)*
                                   6+XX,2,0))
            DO INI_5 WITH "PL"+Y,
                         Z_STR[PL],A_PL&X
            HD1=HD1+HDS+STR((R-1)*
                                   6+XX,2,0)
        NEXT
        @L+5,3 SAY "[ REGION: "+
                  RTRIM(SUBSTR(C[PL],6))+",
                    YEAR: "+STR(Y_B,4,0)+" ]"
        @L+7,3 SAY HD1
        @L+8,3 SAY "------------------ -------
```

```
                  ------- ------- ------- ------- -------"
@L+9,3 SAY  "Productive area(ha):"
FOR XX=1 TO 6
    X=STR(XX,1,0)
    @L+9,15+9*XX SAY
          A_PL&X[1] PICTURE "9999999"

NEXT
FOR Z=2 TO 12
    DO CASE
    CASE Z=2
        @L+10,3 SAY
                      "MAI      (m3/ha/yr):"
    CASE Z=3
        @L+11,3 SAY
                      "Age            (yr):"
    CASE Z=4
        @L+12,3 SAY
                      "Crop density       :"
    CASE Z=5
        @L+13,3 SAY
                      "Working Cycle. (yr):"
    CASE Z=6
        @L+15,3 SAY
                      "Yield rate  (m3/ha):"
    CASE Z=7
        @L+17,3 SAY
                        "Yield distribution"
        @L+18,3 SAY
                      "    - Timber   (%):"
    CASE Z=8
        @L+19,3 SAY
                      "    - Pole     (%):"
    CASE Z=9
        @L+20,3 SAY
                      "    - Fuel     (%):"
    CASE Z=10
        @L+21,3 SAY
                      "    - Pulp     (%):"
    CASE Z=11
        @L+23,3 SAY
                             "Increase rate"
        @L+24,3 SAY
                      " -Prod. area (%/yr):"
    CASE Z=12
        @L+25,3 SAY
                      " -Yield      (%/yr):"
    ENDCASE
    FOR XX=1 TO 6
        X=STR(XX,1,0)
        @PROW(),15+9*XX SAY
          A_PL&X[Z] PICTURE "999.999"
    NEXT
  NEXT
  IF L=0
      L=28
  ELSE
      L=0
      @2,3 SAY HD
  ENDIF
NEXT
NEXT
@2,3 SAY HD
@5,3 SAY "INITIAL INFORMATION:
                          YEAR "+STR(Y_B,4,0)
@7,3 SAY "  VILLAGE FOREST  >  N.WEST
                    N.CEN.  WEST  SOUTH
@8,3 SAY "------------------ S.EAST N.EAST  CHT"
                ------ ------ ------
@9,3 SAY  "YIELD"     ------ ------ ------ ------"
@10,3 SAY "-----"
```

```
FOR X=1 TO 6
    DO CASE
    CASE X=1
        @11,3 SAY "-Timber volume.(m3):"
    CASE X=2
        @13,3 SAY "-Pole volume ..(m3):"
    CASE X=3
        @15,3 SAY "-Fuelwood vol. (m3):"
    CASE X=4
        @17,3 SAY "GROWTH RATE"
        @18,3 SAY "-----------"
        @19,3 SAY "  -Timber   (%/yr):"
    CASE X=5
        @21,3 SAY "  -Pole     (%/yr):"
    CASE X=6
        @23,3 SAY "  -Fuelwood (%/yr):"
    ENDCASE
    Y=LTRIM(STR(X,2,0))
    FOR Z=1 TO 7
        IF X<4
            @PROW(),15+8*Z SAY
                    A_V&Y[Z] PICTURE "99999.9"
        ELSE
            @PROW(),15+8*Z SAY
                    A_V&Y[Z] PICTURE "999.999"
        ENDIF
    NEXT
NEXT
@64,3 SAY "Date: "+DTOC(DATE())
@0,0 SAY " "
SET DEVICE TO SCREEN
RETURN
*!*************************************
*!   Procedure: S_PR2X
*!   Called by: S_PR2         (procedure in SIM.PRG)
*!   Calls: I_TOT             (procedure in SIM.PRG)
*!*************************************
PROC S_PR2X
DO I_TOT
IF X>45
    X=0
    @X+2,3 SAY HD
ENDIF
@X+4,3 SAY HD3
@X+5,3 SAY HD4
@X+5,66 SAY "(in '000 m3)"
@X+7,3 SAY HD1
@X+8,3 SAY HD2
X=X+9
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FLD="Y"+STR(I,4,0)
    FOR V=1 TO 8
        IF V#8
            SEEK A_RC2[V]
            IF V=1
                @X,3 SAY
                        "YEAR-"+STR(I,4,0)

            ENDIF
            @X,6+8*V SAY &FLD
                        PICTURE "9999.99"

            TOT[(I-Y_B)/Y_I+1]=
                   TOT[(I-Y_B)/Y_I+1]+&FLD

        ELSE
            @X,6+8*V SAY TOT[(I-Y_B)/
                   Y_I+1] PICTURE "99999.99"

        ENDIF
    NEXT
    X=X+2
    IF X=57.AND.I<Y_B+Y_N+Y_I
        @2,3 SAY HD
        @4,3 SAY HD3
```

```
            @5,3 SAY HD4
            @5,66 SAY "(in '000 m3)"
            @7,3 SAY HD1
            @8,3 SAY HD2
            X=9
        ENDIF
NEXT
RETURN
*!*************************************
*!   Procedure: S_PR2
*!   Called by: S_PR          (procedure in SIM.PRG)
*!   Calls: S_PR2X            (procedure in SIM.PRG)
*!   Uses: &F_NAME2
*!   Indexes: &F_NAME2
*!*************************************
PROC S_PR2
SELE A
USE &F_NAME2 INDEX &F_NAME2
HD='FUTURE PROJECTIONS FROM
                    SUPPLY SIMULATION'
HD1="        N.WEST N.CENTR.  WEST
                 SOUTH   S.EAST  N.EAST
                           CHT   NATIONAL"
HD2="        ------- ------- ------- ------- -------
                        ------- ------- --------"
DECLARE A_RC2[7]
DECLARE TOT[Y_N+1]
A_RC2[1]="NF_T_NW"
A_RC2[2]="NF_T_NC"
A_RC2[3]="NF_T_W"
A_RC2[4]="NF_T_S"
A_RC2[5]="NF_T_SE"
A_RC2[6]="NF_T_NE"
A_RC2[7]="NF_T_CHT"
SET DEVICE TO PRINT
HD3="TIMBER - NATURAL FOREST"
HD4="-----------------------"
X=50
DO S_PR2X
A_RC2[1]="PL_T_NW"
A_RC2[2]="PL_T_NC"
A_RC2[3]="PL_T_W"
A_RC2[4]="PL_T_S"
A_RC2[5]="PL_T_SE"
A_RC2[6]="PL_T_NE"
A_RC2[7]="PL_T_CHT"
HD3="TIMBER - PLANTATION"
HD4="-------------------"
DO S_PR2X
A_RC2[1]="VF_T_NW"
A_RC2[2]="VF_T_NC"
A_RC2[3]="VF_T_W"
A_RC2[4]="VF_T_S"
A_RC2[5]="VF_T_SE"
A_RC2[6]="VF_T_NE"
A_RC2[7]="VF_T_CHT"
HD3="TIMBER - VILLAGE FOREST"
HD4="-----------------------"
DO S_PR2X
A_RC2[1]="NF_P_NW"
A_RC2[2]="NF_P_NC"
A_RC2[3]="NF_P_W"
A_RC2[4]="NF_P_S"
A_RC2[5]="NF_P_SE"
A_RC2[6]="NF_P_NE"
A_RC2[7]="NF_P_CHT"
HD3="POLE - NATURAL FOREST"
HD4="---------------------"
DO S_PR2X
A_RC2[1]="PL_P_NW"
A_RC2[2]="PL_P_NC"
```

```
A_RC2[3]="PL_P_W"
A_RC2[4]="PL_P_S"
A_RC2[5]="PL_P_SE"
A_RC2[6]="PL_P_NE"
A_RC2[7]="PL_P_CHT"
HD3="POLE - PLANTATION"
HD4="-----------------"
DO S_PR2X
A_RC2[1]="VF_P_NW"
A_RC2[2]="VF_P_NC"
A_RC2[3]="VF_P_W"
A_RC2[4]="VF_P_S"
A_RC2[5]="VF_P_SE"
A_RC2[6]="VF_P_NE"
A_RC2[7]="VF_P_CHT"
HD3="POLE - VILLAGE FOREST"
HD4="---------------------"
DO S_PR2X
A_RC2[1]="NF_F_NW"
A_RC2[2]="NF_F_NC"
A_RC2[3]="NF_F_W"
A_RC2[4]="NF_F_S"
A_RC2[5]="NF_F_SE"
A_RC2[6]="NF_F_NE"
A_RC2[7]="NF_F_CHT"
HD3="FUELWOOD - NATURAL FOREST"
HD4="-------------------------"
DO S_PR2X
A_RC2[1]="PL_F_NW"
A_RC2[2]="PL_F_NC"
A_RC2[3]="PL_F_W"
A_RC2[4]="PL_F_S"
A_RC2[5]="PL_F_SE"
A_RC2[6]="PL_F_NE"
A_RC2[7]="PL_F_CHT"
HD3="FUELWOOD - PLANTATION"
HD4="---------------------"
DO S_PR2X
A_RC2[1]="VF_F_NW"
A_RC2[2]="VF_F_NC"
A_RC2[3]="VF_F_W"
A_RC2[4]="VF_F_S"
A_RC2[5]="VF_F_SE"
A_RC2[6]="VF_F_NE"
A_RC2[7]="VF_F_CHT"
HD3="FUELWOOD - VILLAGE FOREST"
HD4="-------------------------"
DO S_PR2X
A_RC2[1]="NF_PW_NW"
A_RC2[2]="NF_PW_NC"
A_RC2[3]="NF_PW_W"
A_RC2[4]="NF_PW_S"
A_RC2[5]="NF_PW_SE"
A_RC2[6]="NF_PW_NE"
A_RC2[7]="NF_PW_CHT"
HD3="PULPWOOD - NATURAL FOREST"
HD4="-------------------------"
DO S_PR2X
A_RC2[1]="PL_PW_NW"
A_RC2[2]="PL_PW_NC"
A_RC2[3]="PL_PW_W"
A_RC2[4]="PL_PW_S"
A_RC2[5]="PL_PW_SE"
A_RC2[6]="PL_PW_NE"
A_RC2[7]="PL_PW_CHT"
HD3="PULPWOOD - PLANTATION"
HD4="---------------------"
DO S_PR2X
@64,3 SAY "Date: "+DTOC(DATE())
@0,0 SAY " "
SET DEVICE TO SCREEN
```

```
RETURN
*!****************************************
*!     Procedure: B_PR
*!     Called by: BALANCE              (proc in SIM.PRG)
*!     Calls
*!          : I_T8    (procedure in SIM.PRG)
*!          : I_TY    (procedure in SIM.PRG)
*!     Uses: &F_NAME2
*!     Indexes: &F_NAME2
*!****************************************
PROC B_PR
SELE A
USE &F_NAME2 INDEX &F_NAME2
HD='BALANCE FROM FUTURE
                                PROJECTIONS'

HD1="           N.WEST N.CENTR.  WEST
                      SOUTH   S.EAST N.EAST
                              CHT   NATIONAL"

HD2="           ------- ------- ------- ------- -------
                              ------- ------- -------"

DECLARE A_S1[7]
DECLARE A_S2[7]
DECLARE A_S3[7]
DECLARE A_S4[7]
DECLARE A_S5[7]
DECLARE A_D_T[8]
DECLARE A_S_T[8]
DECLARE A_DOT[Y_N+1]
DECLARE A_INT[Y_N+1]
DECLARE A_NFT[Y_N+1]
DECLARE A_PLT[Y_N+1]
DECLARE A_VFT[Y_N+1]
A_S1[1]="NW_FWD_D"
A_S1[2]="NC_FWD_D"
A_S1[3]="W_FWD_D"
A_S1[4]="S_FWD_D"
A_S1[5]="SE_FWD_D"
A_S1[6]="NE_FWD_D"
A_S1[7]="CHT_FWD_D"
A_S2[1]="NW_FWD_I"
A_S2[2]="NC_FWD_I"
A_S2[3]="W_FWD_I"
A_S2[4]="S_FWD_I"
A_S2[5]="SE_FWD_I"
A_S2[6]="NE_FWD_I"
A_S2[7]="CHT_FWD_I"
A_S3[1]="NF_F_NW"
A_S3[2]="NF_F_NC"
A_S3[3]="NF_F_W"
A_S3[4]="NF_F_S"
A_S3[5]="NF_F_SE"
A_S3[6]="NF_F_NE"
A_S3[7]="NF_F_CHT"
A_S4[1]="PL_F_NW"
A_S4[2]="PL_F_NC"
A_S4[3]="PL_F_W"
A_S4[4]="PL_F_S"
A_S4[5]="PL_F_SE"
A_S4[6]="PL_F_NE"
A_S4[7]="PL_F_CHT"
A_S5[1]="VF_F_NW"
A_S5[2]="VF_F_NC"
A_S5[3]="VF_F_W"
A_S5[4]="VF_F_S"
A_S5[5]="VF_F_SE"
A_S5[6]="VF_F_NE"
A_S5[7]="VF_F_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
SET DEVICE TO PRINT
@2,3 SAY HD
@4,3 SAY   "FUELWOOD"
@4,66 SAY "(in '000 m3)"
@5,3 SAY  "--------"
@6,3 SAY HD1
@7,3 SAY HD2
X=9
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FLD="Y"+STR(I,4,0)
    @X-1,3 SAY   "YEAR: "+STR(I,4,0)
    @X,3 SAY "----------"
    @X+1,3 SAY "DEMAND"
    @X+2,3 SAY "-DOMESTIC"
    FOR V=1 TO 8
        IF V# 8
            SEEK A_S1[V]
            @PROW(),6+8*V SAY &FLD
                              PICTURE "9999999"
            A_DOT[(I-Y_B)/ Y_I+1]=
                  A_DOT[(I-Y_B)/ Y_I+1]+&FLD
            A_D_T[V]=A_D_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_DOT[(I-Y_B)/ Y_I+1]
                              PICTURE "99999999"

        ENDIF
    NEXT
    @X+3,3 SAY "-INDUSTRIAL"
    FOR V=1 TO 8
        IF V# 8
            SEEK A_S2[V]
            @PROW(),6+8*V SAY &FLD
                              PICTURE "9999999"
            A_INT[(I-Y_B)/ Y_I+1]=
                  A_INT[(I-Y_B)/ Y_I+1]+&FLD
            A_D_T[V]=A_D_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_INT[(I-Y_B)/ Y_I+1]
                              PICTURE "99999999"

        ENDIF
    NEXT
    @X+4,3 SAY HD2
    @X+5,3 SAY "TOTAL"
    FOR V=1 TO 8
        IF V# 8
            @PROW(),6+8*V SAY
                  A_D_T[V] PICTURE "9999999"
        ELSE
            A_D_T[V]=A_DOT[(I-Y_B)/
                  Y_I+1]+A_INT[(I-Y_B)/ Y_I+1]
            @PROW(),6+8*V SAY
                  A_D_T[V] PICTURE "99999999"
        ENDIF
    NEXT
    @X+6,3 SAY HD2
    @X+7,3 SAY "SUPPLY"
    @X+8,3 SAY "-NAT.FOREST"
    FOR V=1 TO 8
        IF V# 8
            SEEK A_S3[V]
            @PROW(),6+8*V SAY &FLD
                              PICTURE "9999999"
            A_NFT[(I-Y_B)/ Y_I+1]=
                  A_NFT[(I-Y_B)/ Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_NFT[(I-Y_B)/ Y_I+1]
                              PICTURE "99999999"

        ENDIF
    NEXT
    @X+9,3 SAY "-PLANTATION"
    FOR V=1 TO 8
        IF V# 8
            SEEK A_S4[V]
            @PROW(),6+8*V SAY &FLD
                              PICTURE "9999999"
            A_PLT[(I-Y_B)/ Y_I+1]=
                  A_PLT[(I-Y_B)/ Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_PLT[(I-Y_B)/ Y_I+1]
                              PICTURE "99999999"

        ENDIF
    NEXT
    @X+10,3 SAY "-VIL.FOREST"
    FOR V=1 TO 8
        IF V# 8
            SEEK A_S5[V]
            @PROW(),6+8*V SAY &FLD
                              PICTURE "9999999"
            A_VFT[(I-Y_B)/ Y_I+1]=
                  A_VFT[(I-Y_B)/ Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_VFT[(I-Y_B)/ Y_I+1]
                              PICTURE "99999999"

        ENDIF
    NEXT
    @X+11,3 SAY HD2
    @X+12,3 SAY "TOTAL"
    FOR V=1 TO 8
        IF V# 8
            @PROW(),6+8*V SAY
                  A_S_T[V] PICTURE "9999999"
        ELSE
            A_S_T[V]=A_NFT[(I-Y_B)/
                Y_I+1]+A_PLT[(I-Y_B)/ Y_I+1]+
                   A_VFT[(I-Y_B)/ Y_I+1]
            @PROW(),6+8*V SAY
                  A_S_T[V] PICTURE "99999999"
        ENDIF
    NEXT
    @X+13,3 SAY HD2
    @X+14,3 SAY "BALANCE"
    FOR V=1 TO 8
        IF V# 8
            @PROW(),6+8*V SAY
                           A_S_T[V]-A_D_T[V]
                            PICTURE "9999999"
        ELSE
            @PROW(),6+8*V SAY
                           A_S_T[V]-A_D_T[V]
                          PICTURE "99999999"

        ENDIF
    NEXT
    @X+15,3 SAY HD2
    DO CASE
    CASE X=9
        X=27
    CASE X=27
        X=45
    CASE I<Y_B+Y_N
        @2,3 SAY HD
```

```
              @4,3 SAY  "FUELWOOD"
              @4,66 SAY "(in '000 m3)"
              @5,3 SAY  "--------"
              @6,3 SAY HD1
              @7,3 SAY HD2
              X=9
       ENDCASE
   NEXT
   A_S1[1]="NW_SW"
   A_S1[2]="NC_SW"
   A_S1[3]="W_SW"
   A_S1[4]="S_SW"
   A_S1[5]="SE_SW"
   A_S1[6]="NE_SW"
   A_S1[7]="CHT_SW"
   A_S2[1]="NW_PANEL"
   A_S2[2]="NC_PANEL"
   A_S2[3]="W_PANEL"
   A_S2[4]="S_PANEL"
   A_S2[5]="SE_PANEL"
   A_S2[6]="NE_PANEL"
   A_S2[7]="CHT_PANEL"
   A_S3[1]="NF_T_NW"
   A_S3[2]="NF_T_NC"
   A_S3[3]="NF_T_W"
   A_S3[4]="NF_T_S"
   A_S3[5]="NF_T_SE"
   A_S3[6]="NF_T_NE"
   A_S3[7]="NF_T_CHT"
   A_S4[1]="PL_T_NW"
   A_S4[2]="PL_T_NC"
   A_S4[3]="PL_T_W"
   A_S4[4]="PL_T_S"
   A_S4[5]="PL_T_SE"
   A_S4[6]="PL_T_NE"
   A_S4[7]="PL_T_CHT"
   A_S5[1]="VF_T_NW"
   A_S5[2]="VF_T_NC"
   A_S5[3]="VF_T_W"
   A_S5[4]="VF_T_S"
   A_S5[5]="VF_T_SE"
   A_S5[6]="VF_T_NE"
   A_S5[7]="VF_T_CHT"
   DO I_T8 WITH A_D_T
   DO I_T8 WITH A_S_T
   DO I_TY WITH A_DOT
   DO I_TY WITH A_INT
   DO I_TY WITH A_NFT
   DO I_TY WITH A_PLT
   DO I_TY WITH A_VFT
   @2,3 SAY HD
   @4,3 SAY  "TIMBER"
   @4,66 SAY "(in '000 m3)"
   @5,3 SAY  "------"
   @6,3 SAY HD1
   @7,3 SAY HD2
   X=9
   FOR I=Y_B TO Y_B+Y_N STEP Y_I
       FLD="Y"+STR(I,4,0)
       @X-1,3 SAY "YEAR: "+STR(I,4,0)
       @X,3 SAY "----------"
       @X+1,3 SAY "DEMAND"
       @X+2,3 SAY  "-DOM.+COMM."
       FOR V=1 TO 8
           IF V#8
               SEEK A_S1[V]
               @PROW(),6+8*V SAY &FLD
                               PICTURE "9999999"
               A_DOT[(I-Y_B)/Y_I+1]=
                     A_DOT[(I-Y_B)/Y_I+1]+&FLD
               A_D_T[V]=A_D_T[V]+&FLD
```

```
           ELSE
               @PROW(),6+8*V SAY
                                   A_DOT[(I-Y_B)/Y_I+1]
                                     PICTURE "99999999"
           ENDIF
       NEXT
       @X+3,3 SAY  "-PANEL PR."
       FOR V=1 TO 8
           IF V#8
               SEEK A_S2[V]
               @PROW(),6+8*V SAY &FLD
                                   PICTURE "9999999"
               A_INT[(I-Y_B)/Y_I+1]=
                     A_INT[(I-Y_B)/Y_I+1]+&FLD
               A_D_T[V]=A_D_T[V]+&FLD
           ELSE
               @PROW(),6+8*V SAY
                                   A_INT[(I-Y_B)/Y_I+1]
                                     PICTURE "99999999"
           ENDIF
       NEXT
       @X+4,3 SAY HD2
       @X+5,3 SAY "TOTAL"
       FOR V=1 TO 8
           IF V#8
               @PROW(),6+8*V SAY
                       A_D_T[V] PICTURE "9999999"
           ELSE
               A_D_T[V]=A_DOT[(I-Y_B)/
                     Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
               @PROW(),6+8*V SAY
                       A_D_T[V] PICTURE "99999999"
           ENDIF
       NEXT
       @X+6,3 SAY HD2
       @X+7,3 SAY "SUPPLY"
       @X+8,3 SAY "-NAT.FOREST"
       FOR V=1 TO 8
           IF V#8
               SEEK A_S3[V]
               @PROW(),6+8*V SAY &FLD
                                   PICTURE "9999999"
               A_NFT[(I-Y_B)/Y_I+1]=
                     A_NFT[(I-Y_B)/Y_I+1]+&FLD
               A_S_T[V]=A_S_T[V]+&FLD
           ELSE
               @PROW(),6+8*V SAY
                                 A_NFT[(I-Y_B)/Y_I+1]
                                     PICTURE "99999999"
           ENDIF
       NEXT
       @X+9,3 SAY "-PLANTATION"
       FOR V=1 TO 8
           IF V#8
               SEEK A_S4[V]
               @PROW(),6+8*V SAY &FLD
                                   PICTURE "9999999"
               A_PLT[(I-Y_B)/Y_I+1]=
                     A_PLT[(I-Y_B)/Y_I+1]+&FLD
               A_S_T[V]=A_S_T[V]+&FLD
           ELSE
               @PROW(),6+8*V SAY
                                 A_PLT[(I-Y_B)/Y_I+1]
                                     PICTURE "99999999"
           ENDIF
       NEXT
       @X+10,3 SAY "-VIL.FOREST"
       FOR V=1 TO 8
           IF V#8
               SEEK A_S5[V]
```

```
          @PROW(),6+8*V SAY &FLD
                            PICTURE "9999999"
          A_VFT[(I-Y_B)/Y_I+1]=
                A_VFT[(I-Y_B)/Y_I+1]+&FLD
          A_S_T[V]=A_S_T[V]+&FLD
       ELSE
          @PROW(),6+8*V SAY
                         A_VFT[(I-Y_B)/Y_I+1]
                             PICTURE "99999999"
       ENDIF
    NEXT
    @X+11,3 SAY HD2
    @X+12,3 SAY "TOTAL"
    FOR V=1 TO 8
       IF V#8
          @PROW(),6+8*V SAY A_S_T[V]
                            PICTURE "9999999"
       ELSE
          A_S_T[V]=A_NFT[(I-Y_B)/
             Y_I+1]+A_PLT[(I-Y_B)/Y_I+1]+
                        A_VFT[(I-Y_B)/Y_I+1]
          @PROW(),6+8*V SAY A_S_T[V]
                           PICTURE "99999999"
       ENDIF
    NEXT
    @X+13,3 SAY HD2
    @X+14,3 SAY "BALANCE"
    FOR V=1 TO 8
       IF V#8
          @PROW(),6+8*V SAY
                          A_S_T[V]-A_D_T[V]
                            PICTURE "9999999"
       ELSE
          @PROW(),6+8*V SAY
                          A_S_T[V]-A_D_T[V]
                           PICTURE "99999999"
       ENDIF
    NEXT
    @X+15,3 SAY HD2
    DO CASE
    CASE X=9
       X=27
    CASE X=27
       X=45
    CASE I<Y_B+Y_N
       @2,3 SAY HD
       @4,3 SAY  "TIMBER"
       @4,66 SAY "(in '000 m3)"
       @5,3 SAY  "------"
       @6,3 SAY HD1
       @7,3 SAY HD2
       X=9
    ENDCASE
NEXT
A_S1[1]="NW_PT_D"
A_S1[2]="NC_PT_D"
A_S1[3]="W_PT_D"
A_S1[4]="S_PT_D"
A_S1[5]="SE_PT_D"
A_S1[6]="NE_PT_D"
A_S1[7]="CHT_PT_D"
A_S2[1]="NW_PT_I"
A_S2[2]="NC_PT_I"
A_S2[3]="W_PT_I"
A_S2[4]="S_PT_I"
A_S2[5]="SE_PT_I"
A_S2[6]="NE_PT_I"
A_S2[7]="CHT_PT_I"
A_S3[1]="NF_P_NW"
A_S3[2]="NF_P_NC"
A_S3[3]="NF_P_W"
A_S3[4]="NF_P_S"
A_S3[5]="NF_P_SE"
A_S3[6]="NF_P_NE"
A_S3[7]="NF_P_CHT"
A_S4[1]="PL_P_NW"
A_S4[2]="PL_P_NC"
A_S4[3]="PL_P_W"
A_S4[4]="PL_P_S"
A_S4[5]="PL_P_SE"
A_S4[6]="PL_P_NE"
A_S4[7]="PL_P_CHT"
A_S5[1]="VF_P_NW"
A_S5[2]="VF_P_NC"
A_S5[3]="VF_P_W"
A_S5[4]="VF_P_S"
A_S5[5]="VF_P_SE"
A_S5[6]="VF_P_NE"
A_S5[7]="VF_P_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY HD
@4,3 SAY  "POST & POLES"
@4,66 SAY "(in '000 m3)"
@5,3 SAY  "------------"
@6,3 SAY HD1
@7,3 SAY HD2
X=9
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FLD="Y"+STR(I,4,0)
    @X-1,3 SAY "YEAR: "+STR(I,4,0)
    @X,3 SAY   "----------"
    @X+1,3 SAY "DEMAND"
    @X+2,3 SAY "-DOMESTIC"
    FOR V=1 TO 8
       IF V#8
          SEEK A_S1[V]
          @PROW(),6+8*V SAY &FLD
                            PICTURE "9999999"
          A_DOT[(I-Y_B)/Y_I+1]=
                A_DOT[(I-Y_B)/Y_I+1]+&FLD
          A_D_T[V]=A_D_T[V]+&FLD
       ELSE
          @PROW(),6+8*V SAY
                         A_DOT[(I-Y_B)/Y_I+1]
                             PICTURE "99999999"
       ENDIF
    NEXT
    @X+3,3 SAY "-INDUSTRIAL"
    FOR V=1 TO 8
       IF V#8
          SEEK A_S2[V]
          @PROW(),6+8*V SAY &FLD
                            PICTURE "9999999"
          A_INT[(I-Y_B)/Y_I+1]=
                A_INT[(I-Y_B)/Y_I+1]+&FLD
          A_D_T[V]=A_D_T[V]+&FLD
       ELSE
          @PROW(),6+8*V SAY
                         A_INT[(I-Y_B)/Y_I+1]
                             PICTURE "99999999"
       ENDIF
    NEXT
    @X+4,3 SAY HD2
    @X+5,3 SAY "TOTAL"
    FOR V=1 TO 8
       IF V#8
```

```
        @PROW(),6+8*V SAY
                 A_D_T[V] PICTURE "9999999"
    ELSE
        A_D_T[V]=A_DOT[(I-Y_B)/
              Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
        @PROW(),6+8*V SAY
                 A_D_T[V] PICTURE "99999999"
    ENDIF
NEXT
@X+6,3 SAY HD2
@X+7,3 SAY "SUPPLY"
@X+8,3 SAY "-NAT.FOREST"
FOR V=1 TO 8
    IF V#8
        SEEK A_S3[V]
        @PROW(),6+8*V SAY &FLD
                          PICTURE "9999999"
        A_NFT[(I-Y_B)/Y_I+1]=
              A_NFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
    ELSE
        @PROW(),6+8*V SAY
                      A_NFT[(I-Y_B)/Y_I+1]
                          PICTURE "99999999"
    ENDIF
NEXT
@X+9,3 SAY "-PLANTATION"
FOR V=1 TO 8
    IF V#8
        SEEK A_S4[V]
        @PROW(),6+8*V SAY &FLD
                          PICTURE "9999999"
        A_PLT[(I-Y_B)/Y_I+1]=
              A_PLT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
    ELSE
        @PROW(),6+8*V SAY
                      A_PLT[(I-Y_B)/Y_I+1]
                          PICTURE "99999999"
    ENDIF
NEXT
@X+10,3 SAY "-VIL.FOREST"
FOR V=1 TO 8
    IF V#8
        SEEK A_S5[V]
        @PROW(),6+8*V SAY &FLD
                          PICTURE "9999999"
        A_VFT[(I-Y_B)/Y_I+1]=
              A_VFT[(I-Y_B)/Y_I+1]+&FLD
        A_S_T[V]=A_S_T[V]+&FLD
    ELSE
        @PROW(),6+8*V SAY
                      A_VFT[(I-Y_B)/Y_I+1]
                          PICTURE "99999999"
    ENDIF
NEXT
@X+11,3 SAY HD2
@X+12,3 SAY "TOTAL"
FOR V=1 TO 8
    IF V#8
        @PROW(),6+8*V SAY A_S_T[V]
                          PICTURE "9999999"
    ELSE
        A_S_T[V]=A_NFT[(I-Y_B)/
              Y_I+1]+A_PLT[(I-Y_B)/Y_I+1]
                    +A_VFT[(I-Y_B)/Y_I+1]
        @PROW(),6+8*V SAY A_S_T[V]
                          PICTURE "99999999"
    ENDIF
NEXT
@X+13,3 SAY HD2
```

```
    @X+14,3 SAY "BALANCE"
    FOR V=1 TO 8
        IF V#8
            @PROW(),6+8*V SAY
                                 A_S_T[V]-A_D_T[V]
                                 PICTURE "9999999"
        ELSE
            @PROW(),6+8*V SAY
                                 A_S_T[V]-A_D_T[V]
                                 PICTURE "99999999"
        ENDIF
    NEXT
    @X+15,3 SAY HD2
    DO CASE
    CASE X=9
        X=27
    CASE X=27
        X=45
    CASE I<Y_B+Y_N
        @2,3 SAY HD
        @4,3 SAY  "POST & POLES"
        @4,66 SAY "(in '000 m3)"
        @5,3 SAY  "-----------"
        @6,3 SAY HD1
        @7,3 SAY HD2
        X=9
    ENDCASE
NEXT
A_S1[1]="NW_NP"
A_S1[2]="NC_NP"
A_S1[3]="W_NP"
A_S1[4]="S_NP"
A_S1[5]="SE_NP"
A_S1[6]="NE_NP"
A_S1[7]="CHT_NP"
A_S2[1]="NW_PPR"
A_S2[2]="NC_PPR"
A_S2[3]="W_PPR"
A_S2[4]="S_PPR"
A_S2[5]="SE_PPR"
A_S2[6]="NE_PPR"
A_S2[7]="CHT_PPR"
A_S3[1]="NF_PW_NW"
A_S3[2]="NF_PW_NC"
A_S3[3]="NF_PW_W"
A_S3[4]="NF_PW_S"
A_S3[5]="NF_PW_SE"
A_S3[6]="NF_PW_NE"
A_S3[7]="NF_PW_CHT"
A_S4[1]="PL_PW_NW"
A_S4[2]="PL_PW_NC"
A_S4[3]="PL_PW_W"
A_S4[4]="PL_PW_S"
A_S4[5]="PL_PW_SE"
A_S4[6]="PL_PW_NE"
A_S4[7]="PL_PW_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY HD
@4,3 SAY  "PULPWOOD - POPULATION
            GROWTH WITH CONSTANT LITERACY"
@4,66 SAY "(in '000 m3)"
@5,3 SAY  "--------------------------------
                                   ---------------"
@6,3 SAY HD1
@7,3 SAY HD2
```

```
X=9
FOR I=Y_B TO Y_B+Y_N STEP Y_I
   FLD="Y"+STR(I,4,0)
   @X-1,3 SAY "YEAR: "+STR(I,4,0)
   @X,3 SAY  "----------"
   @X+1,3 SAY "DEMAND"
   @X+2,3 SAY  "-NEWSPRINT"
   FOR V=1 TO 8
        IF V=1
             SEEK A_S1[V]
             A_DOT[(I-Y_B)/Y_I+1]=
                              A_DOT[(I-Y_B)/Y_I+1]
                                       +&FLD*50/35.3
             A_D_T[V]=A_D_T[V]+
                                       &FLD*50/35.3
        ENDIF
        IF V=8
             @PROW(),6+8*V SAY
                              A_DOT[(I-Y_B)/Y_I+1]
                                    PICTURE "99999999"
        ENDIF
   NEXT
   @X+3,3 SAY "-OTHERS"
   FOR V=1 TO 8
        IF V=1
             SEEK A_S2[V]
             A_INT[(I-Y_B)/Y_I+1]=
                              A_INT[(I-Y_B)/Y_I+1]
                                       +&FLD*50/35.3
             A_D_T[V]=A_D_T[V]+&FLD
                                            *50/35.3
        ENDIF
        IF V=8
             @PROW(),6+8*V SAY
                              A_INT[(I-Y_B)/Y_I+1]
                                    PICTURE "99999999"
        ENDIF
   NEXT
   @X+4,3 SAY HD2
   @X+5,3 SAY "TOTAL"
   FOR V=1 TO 8
        IF V# 8
             @PROW(),6+8*V SAY
                     A_D_T[V] PICTURE "9999999"
        ELSE
             A_D_T[V]=A_DOT[(I-Y_B)/
                     Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
             @PROW(),6+8*V SAY
                     A_D_T[V] PICTURE "99999999"
        ENDIF
   NEXT
   @X+6,3 SAY HD2
   @X+7,3 SAY "SUPPLY"
   @X+8,3 SAY "-NAT.FOREST"
   FOR V=1 TO 8
        IF V# 8
             SEEK A_S3[V]
             @PROW(),6+8*V SAY &FLD
                                    PICTURE "9999999"
             A_NFT[(I-Y_B)/Y_I+1]=
                     A_NFT[(I-Y_B)/Y_I+1]+&FLD
             A_S_T[V]=A_S_T[V]+&FLD
        ELSE
             @PROW(),6+8*V SAY
      A_NFT[(I-Y_B)/Y_I+1] PICTURE "99999999"
        ENDIF
   NEXT
   @X+9,3 SAY "-PLANTATION"
   FOR V=1 TO 8
        IF V# 8
             SEEK A_S4[V]
             @PROW(),6+8*V SAY &FLD
                                    PICTURE "9999999"
             A_PLT[(I-Y_B)/Y_I+1]=
                     A_PLT[(I-Y_B)/Y_I+1]+&FLD
             A_S_T[V]=A_S_T[V]+&FLD
        ELSE
             @PROW(),6+8*V SAY
                              A_PLT[(I-Y_B)/Y_I+1]
                                    PICTURE "99999999"
        ENDIF
   NEXT
   @X+10,3 SAY HD2
   @X+11,3 SAY "TOTAL"
   FOR V=1 TO 8
        IF V# 8
             @PROW(),6+8*V SAY
                     A_S_T[V] PICTURE "9999999"
        ELSE
             A_S_T[V]=A_NFT[(I-Y_B)/
                     Y_I+1]+A_PLT[(I-Y_B)/Y_I+1]
                              +A_VFT[(I-Y_B)/Y_I+1]
             @PROW(),6+8*V SAY
                     A_S_T[V] PICTURE "99999999"
        ENDIF
   NEXT
   @X+12,3 SAY HD2
   @X+13,3 SAY "BALANCE"
   FOR V=1 TO 8
        IF V# 8
             @PROW(),6+8*V SAY
                              A_S_T[V]-A_D_T[V]
                                    PICTURE "9999999"
        ELSE
             @PROW(),6+8*V SAY
                              A_S_T[V]-A_D_T[V]
                                    PICTURE "99999999"
        ENDIF
   NEXT
   @X+14,3 SAY HD2
   DO CASE
   CASE X=9
        X=27
   CASE X=27
        X=45
   CASE I<Y_B+Y_N
        @2,3 SAY HD
        @4,3 SAY  "PULPWOOD -
                  POPULATION GROWTH WITH
                           CONSTANT LITERACY"
        @4,66 SAY "(in '000 m3)"
        @5,3 SAY  "----------------------------
                                 ---------------------"
        @6,3 SAY HD1
        @7,3 SAY HD2
        X=9
   ENDCASE
NEXT
A_S1[1]="NW_NP"
A_S1[2]="NC_NP"
A_S1[3]="W_NP"
A_S1[4]="S_NP"
A_S1[5]="SE_NP"
A_S1[6]="NE_NP"
A_S1[7]="CHT_NP"
A_S2[1]="NW_PPR"
A_S2[2]="NC_PPR"
A_S2[3]="W_PPR"
A_S2[4]="S_PPR"
A_S2[5]="SE_PPR"
A_S2[6]="NE_PPR"
A_S2[7]="CHT_PPR"
A_S3[1]="NF_PW_NW"
A_S3[2]="NF_PW_NC"
A_S3[3]="NF_PW_W"
A_S3[4]="NF_PW_S"
A_S3[5]="NF_PW_SE"
A_S3[6]="NF_PW_NE"
A_S3[7]="NF_PW_CHT"
A_S4[1]="PL_PW_NW"
A_S4[2]="PL_PW_NC"
```

```
A_S4[3]="PL_PW_W"
A_S4[4]="PL_PW_S"
A_S4[5]="PL_PW_SE"
A_S4[6]="PL_PW_NE"
A_S4[7]="PL_PW_CHT"
DO I_T8 WITH A_D_T
DO I_T8 WITH A_S_T
DO I_TY WITH A_DOT
DO I_TY WITH A_INT
DO I_TY WITH A_NFT
DO I_TY WITH A_PLT
DO I_TY WITH A_VFT
@2,3 SAY HD
@4,3 SAY  "PULPWOOD - POPULATION AND
                          LITERACY GROWTH"
@4,66 SAY "(in '000 m3)"
@5,3 SAY  "----------------------------------------"
@6,3 SAY HD1
@7,3 SAY HD2
X=9
FOR I=Y_B TO Y_B+Y_N STEP Y_I
    FLD="Y"+STR(I,4,0)
    @X-1,3 SAY "YEAR: "+STR(I,4,0)
    @X,3 SAY   "----------"
    @X+1,3 SAY "DEMAND"
    @X+2,3 SAY  "-NEWSPRINT"
    FOR V=1 TO 8
        IF V=2
            SEEK A_S1[V]
            A_DOT[(I-Y_B)/Y_I+1]=
                         A_DOT[(I-Y_B)/Y_I+1]
                                 +&FLD*50/35.3
            A_D_T[V]=A_D_T[V]
                                 +&FLD*50/35.3
        ENDIF
        IF V=8
            @PROW(),6+8*V SAY
                         A_DOT[(I-Y_B)/Y_I+1]
                            PICTURE "99999999"
        ENDIF
    NEXT
    @X+3,3 SAY "-OTHERS"
    FOR V=1 TO 8
        IF V=2
            SEEK A_S2[V]
            A_INT[(I-Y_B)/Y_I+1]=
                          A_INT[(I-Y_B)/Y_I+1]
                                 +&FLD*50/35.3
            A_D_T[V]=A_D_T[V]
                                 +&FLD*50/35.3
        ENDIF
        IF V=8
            @PROW(),6+8*V SAY
                          A_INT[(I-Y_B)/Y_I+1]
                            PICTURE "99999999"
        ENDIF
    NEXT
    @X+4,3 SAY HD2
    @X+5,3 SAY "TOTAL"
    FOR V=1 TO 8
        IF V#8
            @PROW(),6+8*V SAY
                    A_D_T[V] PICTURE "9999999"
        ELSE
            A_D_T[V]=A_DOT[(I-Y_B)/
                    Y_I+1]+A_INT[(I-Y_B)/Y_I+1]
            @PROW(),6+8*V SAY
                    A_D_T[V] PICTURE "99999999"
        ENDIF
    NEXT
    @X+6,3 SAY HD2
    @X+7,3 SAY "SUPPLY"
    @X+8,3 SAY "-NAT.FOREST"
    FOR V=1 TO 8
        IF V#8
            SEEK A_S3[V]
            @PROW(),6+8*V SAY &FLD
                               PICTURE "9999999"
            A_NFT[(I-Y_B)/Y_I+1]=
                   A_NFT[(I-Y_B)/Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_NFT[(I-Y_B)/Y_I+1]
                              PICTURE "99999999"
        ENDIF
    NEXT
    @X+9,3 SAY "-PLANTATION"
    FOR V=1 TO 8
        IF V#8
            SEEK A_S4[V]
            @PROW(),6+8*V SAY &FLD
                               PICTURE "9999999"
            A_PLT[(I-Y_B)/Y_I+1]=
                   A_PLT[(I-Y_B)/Y_I+1]+&FLD
            A_S_T[V]=A_S_T[V]+&FLD
        ELSE
            @PROW(),6+8*V SAY
                           A_PLT[(I-Y_B)/Y_I+1]
                              PICTURE "99999999"
        ENDIF
    NEXT
    @X+10,3 SAY HD2
    @X+11,3 SAY "TOTAL"
    FOR V=1 TO 8
        IF V#8
            @PROW(),6+8*V SAY A_S_T[V]
                               PICTURE "9999999"
        ELSE
            A_S_T[V]=A_NFT[(I-Y_B)/
                   Y_I+1]+A_PLT[(I-Y_B)/Y_I+1]
                         +A_VFT[(I-Y_B)/Y_I+1]
            @PROW(),6+8*V SAY A_S_T[V]
                              PICTURE "99999999"
        ENDIF
    NEXT
    @X+12,3 SAY HD2
    @X+13,3 SAY "BALANCE"
    FOR V=1 TO 8
        IF V#8       @PROW(),6+8*V SAY
                               A_S_T[V]-A_D_T[V]
                               PICTURE "9999999"
        ELSE
            @PROW(),6+8*V SAY
                               A_S_T[V]-A_D_T[V]
                              PICTURE "99999999"
        ENDIF
    NEXT
    @X+14,3 SAY HD2
    DO CASE
    CASE X=9
        X=27
    CASE X=27
        X=45
    CASE I<Y_B+Y_N
        @2,3 SAY HD
        @4,3 SAY  "PULPWOOD -
                               POPULATION AND
                           LITERACY GROWTH"
        @4,66 SAY "(in '000 m3)"
        @5,3 SAY  "-------------------
                                    ---------------"
        @6,3 SAY HD1
        @7,3 SAY HD2
        X=9
    ENDCASE
NEXT
@64,3 SAY "Date: "+DTOC(DATE())
@0,0 SAY " "
SET DEVICE TO SCREEN
RETURN
*************************************
```

3. DATA FILE STRUCTURE

Structure for database: IP1.DBF

Field	Field Name	Type	Width	Dec	Remarks
1	ITEM1	Character	20		Item description
2	G_RATE1	Numeric	14	5	Growth rate
3	N_WEST	Numeric	14	5	North West Region
4	N_CENT	Numeric	14	5	North Central Region
5	WEST	Numeric	14	5	West Region
6	SOUTH	Numeric	14	5	South Region
7	S_EAST	Numeric	14	5	South East Region
8	N_EAST	Numeric	14	5	North east Region
9	CHT	Numeric	14	5	Chittagong Hill Tracts Region
10	U_RICH	Numeric	14	5	Urban Rich
11	U_POOR	Numeric	14	5	Urban Poor
12	R_RICH	Numeric	14	5	Rural Rich
13	R_POOR	Numeric	14	5	Rural Poor
** Total **			189		

Structure for database: IP2.DBF

Fld	F.Name	Type	Width	Dec	Remarks
1	ITEM2	Character	20		Item description
2	G_RATE2	Numeric	14	5	Growth Rate
3	Y1993	Numeric	14	5	Data for year 1993
4	Y1998	Numeric	14	5	Data for year 1998
5	Y2003	Numeric	14	5	Data for year 2003
6	Y2008	Numeric	14	5	Data for year 2008
7	Y2013	Numeric	14	5	Data for year 2013
** Total **			105		

These two database files and its contents of the fields are automatically created and adjusted by the programme as required.

4. SAMPLE OUTPUT FORMATS

```
INITIAL PARAMETERS FOR DEMAND SIMULATION

[ CONSUMPTION IN THE YEAR 1993 ]

                  ------PER CAPITA CONSUMPTION------¦GOVT.& COMM.¦CONSUMPTION
                        ¦--- URBAN----¦----RURAL----¦CONSUMPTION ¦GROWTH RATE
                  GROSS ¦ RICH ¦ POOR ¦ RICH ¦ POOR ¦(% of urban)¦(% per yr)
                  ------¦------¦------¦------¦------¦------------¦-----------
FUELWOOD

 ->DOMESTIC   (m3):        0.0650 0.0430 0.0650 0.0430                 0.000

 ->INDUSTRIAL(m3): 0.0264                                              0.000

SAWN TIMBER   (m3):        0.1100 0.0111        0.0111      28.597     0.000

 ->RWE FACTOR (%): 37.500

POSTS & POLES

 ->DOMESTIC   (m3): 0.0019                                             0.000

 ->INDUSTRIAL(m3): 0.0008                                              0.000

NEWSPRINT     (kg): 1.9070                                             0.000

OTHER PAPERS (kg): 0.5602                                              0.000

                     NORTH   NORTH                      SOUTH   NORTH
                      WEST  CENTRAL    WEST    SOUTH     EAST    EAST     CHT
                   ------- ------- ------- ------- ------- ------- -------

PANEL PRODUCTS(m3):   5.340  57.810  66.470   0.000  39.380  25.060  10.130


[ POPULATION ]

YEAR  POPULATION  LITERACY  ---- URBAN POPULATION (% of regional total) ----
       (in '000s)   (%)     N.WEST N.CEN.  WEST  SOUTH  S.EAST N.EAST  CHT
----  ----------  --------  ------ ------ ------ ------ ------ ------ ------
1993    112015     26.442     8.78  26.84  13.85   9.80  21.41  11.62  16.54

1998    122068     28.662     9.55  27.61  14.62  10.57  22.18  12.39  17.31

2003    132122     30.882    10.20  28.26  15.27  11.22  22.83  13.04  17.97

2008    142176     33.102    10.76  28.82  15.83  11.78  23.39  13.60  18.52

2013    152230     35.322    11.24  29.30  16.31  12.26  23.87  14.08  19.00


  ----------- POPULATION DISTRIBUTION (as % of total population) ------
  RICH     N.WEST   N.CENT.   WEST     SOUTH    S.EAST   N.EAST     CHT
 -------  -------  -------  -------  -------  -------  -------  -------
  20.000   24.270   21.900   13.590   10.680   16.520   12.120    0.920
```

Figure 1 - Output of Initial Parameters Entered

INITIAL PARAMETERS FOR DEMAND SIMULATION

[POPULATION ('000s)]

YEAR		NORTH WEST	NORTH CENTRAL	WEST	SOUTH	SOUTH EAST	NORTH EAST	CHT
1993	URBAN-RICH:	477	1317	422	234	792	316	34
	-POOR:	1910	5267	1687	938	3170	1262	136
	RURAL-RICH:	4960	3589	2623	2158	2909	2400	172
	-POOR:	19839	14358	10492	8633	11634	9599	688
1998	URBAN-RICH:	566	1476	485	276	895	367	39
	-POOR:	2263	5905	1940	1102	3578	1466	156
	RURAL-RICH:	5359	3870	2833	2332	3139	2592	186
	-POOR:	21437	15482	11331	9327	12554	10369	743
2003	URBAN-RICH:	654	1635	548	317	997	418	44
	-POOR:	2617	6542	2193	1267	3986	1670	175
	RURAL-RICH:	5759	4152	3043	2505	3369	2785	199
	-POOR:	23036	16606	12171	10022	13475	11140	798
2008	URBAN-RICH:	743	1795	612	358	1099	469	48
	-POOR:	2970	7179	2447	1431	4395	1875	194
	RURAL-RICH:	6159	4433	3253	2679	3599	2978	213
	-POOR:	24635	17730	13010	10717	14395	11911	853
2013	URBAN-RICH:	831	1954	675	399	1201	520	53
	-POOR:	3322	7815	2699	1595	4802	2078	213
	RURAL-RICH:	6559	4714	3463	2853	3829	3170	227
	-POOR:	26235	18856	13851	11412	15316	12682	908

Figure 2 - Output of Population as Calculated from Initial Parameters

FUTURE PROJECTIONS FROM DEMAND SIMULATION

FUELWOOD - DOMESTIC USE

(in '000 m3)

	N.WEST	N.CENTR.	WEST	SOUTH	S.EAST	N.EAST	CHT	NATIONAL
YEAR-1993	1288.61	1162.78	721.56	567.05	877.13	643.51	48.85	5309.49
YEAR-1998	1404.27	1267.14	786.32	617.95	955.85	701.27	53.23	5786.04
YEAR-2003	1519.93	1371.51	851.09	668.85	1034.58	759.03	57.62	6262.60
YEAR-2008	1635.59	1475.87	915.85	719.74	1113.31	816.78	62.00	6739.15
YEAR-2013	1751.25	1580.24	980.61	770.64	1192.03	874.54	66.38	7215.70

FUELWOOD - INDUSTRIAL USE

(in '000 m3)

	N.WEST	N.CENTR.	WEST	SOUTH	S.EAST	N.EAST	CHT	NATIONAL
YEAR-1993	717.71	647.62	401.88	315.83	488.53	358.41	27.21	2957.18
YEAR-1998	782.13	705.75	437.95	344.17	532.37	390.58	29.65	3222.61
YEAR-2003	846.54	763.88	474.02	372.52	576.22	422.75	32.09	3488.03
YEAR-2008	910.96	822.01	510.09	400.87	620.07	454.92	34.53	3753.45
YEAR-2013	975.38	880.13	546.16	429.22	663.92	487.09	36.97	4018.87

SAWN TIMBER (RWE)

(in '000 m3)

	N.WEST	N.CENTR.	WEST	SOUTH	S.EAST	N.EAST	CHT	NATIONAL
YEAR-1993	986.81	1228.46	611.45	443.56	850.02	522.21	43.51	4686.03
YEAR-1998	1092.79	1354.43	676.08	491.03	938.15	577.78	48.07	5178.33
YEAR-2003	1198.70	1480.33	740.67	538.47	1026.25	633.31	52.64	5670.36
YEAR-2008	1304.66	1606.28	805.28	585.93	1114.38	688.86	57.20	6162.59
YEAR-2013	1410.44	1732.08	869.80	633.32	1202.39	744.33	61.76	6654.12

Figure 3 - Sample Output Format of Demand Projections

```
INITIAL PARAMETERS FOR SUPPLY SIMULATION

[ REGION: SOUTH, YEAR: 1993 ]

 NATURAL FOREST >       NF  1    NF  2    NF  3    NF  4    NF  5    NF  6
-------------------    -------  -------  -------  -------  -------  -------
Productive area(ha):    15788    11836     9754     2258        0        0
MAI      (m3/ha/yr):    3.000    3.840    0.890    2.550    0.000    0.000
Age           (yr):    45.000   45.000   45.000   15.000    0.000    0.000
Crop density      :     0.700    0.700    0.700    0.700    0.000    0.000
Working Cycle. (yr):   45.000   45.000   45.000   15.000    0.000    0.000

Yield rate  (m3/ha):   94.500  120.960   28.035   26.775    0.000    0.000

Yield distribution
     - Timber   (%):   70.000   64.000    0.000    0.000    0.000    0.000
     - Pole     (%):    0.000    0.000    0.000    0.000    0.000    0.000
     - Fuel     (%):   30.000   36.000  100.000  100.000    0.000    0.000
     - Pulp     (%):    0.000    0.000    0.000    0.000    0.000    0.000

Increase rate
 -Prod. area (%/yr):    0.000    0.000    0.000    0.000    0.000    0.000
 -Yield      (%/yr):    0.000    0.000    0.000    0.000    0.000    0.000
```

```
[ REGION: SOUTH EAST, YEAR: 1993 ]

 NATURAL FOREST >       NF  1    NF  2    NF  3    NF  4    NF  5    NF  6
-------------------    -------  -------  -------  -------  -------  -------
Productive area(ha):     2749     1534     1000    12790        0        0
MAI      (m3/ha/yr):    2.400    0.630    1.900    0.000    0.000    0.000
Age           (yr):    45.000   45.000   15.000    0.000    0.000    0.000
Crop density      :     0.700    0.700    0.700    0.000    0.000    0.000
Working Cycle. (yr):   45.000   45.000   15.000    0.000    0.000    0.000

Yield rate  (m3/ha):   75.600   19.845   19.950    0.000    0.000    0.000

Yield distribution
     - Timber   (%):   67.000    0.000    0.000    0.000    0.000    0.000
     - Pole     (%):    0.000    0.000    0.000    0.000    0.000    0.000
     - Fuel     (%):   33.000  100.000  100.000    0.000    0.000    0.000
     - Pulp     (%):    0.000    0.000    0.000    0.000    0.000    0.000

Increase rate
 -Prod. area (%/yr):    0.000    0.000    0.000    0.000    0.000    0.000
 -Yield      (%/yr):    0.000    0.000    0.000    0.000    0.000    0.000
```

Figure 4 - Sample Output Format of Supply Parameters Entered for Natural Forests

```
INITIAL PARAMETERS FOR SUPPLY SIMULATION

[ REGION: NORTH CENTRAL, YEAR: 1993 ]

 PLANTATION      >       PL  1     PL  2     PL  3     PL  4     PL  5     PL  6
-------------------    -------   -------   -------   -------   -------   -------
Productive area(ha):      3810         0         0         0         0         0
MAI      (m3/ha/yr):    10.000     0.000     0.000     0.000     0.000     0.000
Age            (yr):    20.000     0.000     0.000     0.000     0.000     0.000
Crop density      :      1.000     0.000     0.000     0.000     0.000     0.000
Working Cycle. (yr):     7.000     0.000     0.000     0.000     0.000     0.000

Yield rate  (m3/ha):   200.000     0.000     0.000     0.000     0.000     0.000

Yield distribution
    - Timber    (%):    30.000     0.000     0.000     0.000     0.000     0.000
    - Pole      (%):    20.000     0.000     0.000     0.000     0.000     0.000
    - Fuel      (%):    50.000     0.000     0.000     0.000     0.000     0.000
    - Pulp      (%):     0.000     0.000     0.000     0.000     0.000     0.000

Increase rate
 -Prod. area (%/yr):     0.000     0.000     0.000     0.000     0.000     0.000
 -Yield      (%/yr):     0.000     0.000     0.000     0.000     0.000     0.000



[ REGION: WEST, YEAR: 1993 ]

 PLANTATION      >       PL  1     PL  2     PL  3     PL  4     PL  5     PL  6
-------------------    -------   -------   -------   -------   -------   -------
Productive area(ha):      1580      1562         0         0         0         0
MAI      (m3/ha/yr):    10.000     7.000     0.000     0.000     0.000     0.000
Age            (yr):    20.000     7.000     0.000     0.000     0.000     0.000
Crop density      :      1.000     0.700     0.000     0.000     0.000     0.000
Working Cycle. (yr):     7.000     5.000     0.000     0.000     0.000     0.000

Yield rate  (m3/ha):   200.000    34.300     0.000     0.000     0.000     0.000

Yield distribution
    - Timber    (%):    30.000     0.000     0.000     0.000     0.000     0.000
    - Pole      (%):    20.000    30.000     0.000     0.000     0.000     0.000
    - Fuel      (%):    50.000    70.000     0.000     0.000     0.000     0.000
    - Pulp      (%):     0.000     0.000     0.000     0.000     0.000     0.000

Increase rate
 -Prod. area (%/yr):     0.000     0.000     0.000     0.000     0.000     0.000
 -Yield      (%/yr):     0.000     0.000     0.000     0.000     0.000     0.000
```

Figure 5 - Sample Output Format of Supply Parameters Entered for Plantations

```
FUTURE PROJECTIONS FROM SUPPLY SIMULATION

TIMBER - NATURAL FOREST
-----------------------
                                                                          (in '000 m3)
            N.WEST N.CENTR.   WEST    SOUTH    S.EAST  N.EAST    CHT    NATIONAL
            ------- ------- ------- ------- ------- ------- ------- --------
YEAR-1993     0.00     0.00  169.02  845.08  888.65 1066.03 1078.41  4047.19
YEAR-1998     0.00     0.00  338.03  845.08  932.22 1069.12 1078.41  4262.87
YEAR-2003     0.00     0.00  507.05  845.08  975.79 1072.22 1078.41  4478.56
YEAR-2008     0.00     0.00  676.07  845.08 1019.37 1075.31 1078.41  4694.24
YEAR-2013     0.00     0.00  845.08  845.08 1062.94 1078.41 1078.41  4909.92


TIMBER - PLANTATION
-------------------
                                                                          (in '000 m3)
            N.WEST N.CENTR.   WEST    SOUTH    S.EAST  N.EAST    CHT    NATIONAL
            ------- ------- ------- ------- ------- ------- ------- --------
YEAR-1993    46.50  271.24  458.87  603.04  801.26 1377.43 1377.43  4935.77
YEAR-1998    93.00  309.99  491.53  616.59  945.30 1377.43 1377.43  5211.26
YEAR-2003   139.50  348.73  524.19  630.13 1089.34 1377.43 1377.43  5486.74
YEAR-2008   186.00  387.47  556.84  643.67 1233.39 1377.43 1377.43  5762.23
YEAR-2013   232.50  426.21  589.50  657.21 1377.43 1377.43 1377.43  6037.71


TIMBER - VILLAGE FOREST
-----------------------
                                                                          (in '000 m3)
            N.WEST N.CENTR.   WEST    SOUTH    S.EAST  N.EAST    CHT    NATIONAL
            ------- ------- ------- ------- ------- ------- ------- --------
YEAR-1993     0.00     0.00    0.00    0.00    0.00    0.00    0.00     0.00
YEAR-1998     0.00     0.00    0.00    0.00    0.00    0.00    0.00     0.00
YEAR-2003     0.00     0.00    0.00    0.00    0.00    0.00    0.00     0.00
YEAR-2008     0.00     0.00    0.00    0.00    0.00    0.00    0.00     0.00
YEAR-2013     0.00     0.00    0.00    0.00    0.00    0.00    0.00     0.00
```

Figure 6 - Sample Output Format of Supply Projections

```
BALANCE FROM FUTURE PROJECTIONS
                                                                        (in '000 m3)
FUELWOOD
--------
              N.WEST N.CENTR.  WEST   SOUTH  S.EAST  N.EAST    CHT  NATIONAL
              ------ -------  ------ ------  ------  ------  ------ --------
YEAR: 1993
----------
DEMAND
-DOMESTIC       1289   1163     722    567     877     644      49     5309
-INDUSTRIAL      718    648     402    316     489     358      27     2957
              ------ -------  ------ ------  ------  ------  ------ --------
TOTAL           2006   1810    1123    883    1366    1002      76     8267
              ------ -------  ------ ------  ------  ------  ------ --------
SUPPLY
-NAT.FOREST        0      0      58    291     323     452     467     1591
-PLANTATION      119    780    1581   1829    2045    2428    2428    11209
-VIL.FOREST      301    339     187    258     354     134       0     1573
              ------ -------  ------ ------  ------  ------  ------ --------
TOTAL            420   1119    1826   2378    2722    3014    2895    14373
              ------ -------  ------ ------  ------  ------  ------ --------
BALANCE        -1587   -691     703   1495    1356    2013    2818     6107
              ------ -------  ------ ------  ------  ------  ------ --------

YEAR: 1998
----------
DEMAND
-DOMESTIC       1404   1267     786    618     956     701      53     5786
-INDUSTRIAL      782    706     438    344     532     391      30     3223
              ------ -------  ------ ------  ------  ------  ------ --------
TOTAL           4193   3783    2348   1845    2854    2094     159     9009
              ------ -------  ------ ------  ------  ------  ------ --------
SUPPLY
-NAT.FOREST        0      0     117    291     354     456     467     1685
-PLANTATION      237    967    1635   1859    2141    2428    2428    11694
-VIL.FOREST      301    339     187    258     354     134       0     1573
              ------ -------  ------ ------  ------  ------  ------ --------
TOTAL            958   2424    3765   4786    5571    6032    5789    14952
              ------ -------  ------ ------  ------  ------  ------ --------
BALANCE        -3235  -1359    1417   2941    2717    3939    5630     5944
              ------ -------  ------ ------  ------  ------  ------ --------

YEAR: 2003
----------
DEMAND
-DOMESTIC       1520   1372     851    669    1035     759      58     6263
-INDUSTRIAL      847    764     474    373     576     423      32     3488
              ------ -------  ------ ------  ------  ------  ------ --------
TOTAL           6559   5919    3673   2886    4465    3276     249     9751
              ------ -------  ------ ------  ------  ------  ------ --------
SUPPLY
-NAT.FOREST        0      0     175    291     386     459     467     1778
-PLANTATION      356   1153    1690   1889    2236    2428    2428    12180
-VIL.FOREST      301    339     187    258     354     134       0     1573
              ------ -------  ------ ------  ------  ------  ------ --------
TOTAL           1615   3917    5816   7224    8547    9054    8684    15531
              ------ -------  ------ ------  ------  ------  ------ --------
BALANCE        -4944  -2002    2144   4338    4082    5778    8435     5780
              ------ -------  ------ ------  ------  ------  ------ --------
```

Figure 7 - Sample Output Format of Demand-Supply Balance

(Proj. 372001/25, App. 3)

APPENDIX 4
INVENTORY PROGRAMME STRUCTURE AND CODE

PROJECT 372001/25
FORESTRY MASTER PLAN
BANGLADESH TA 1355-BAN

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 05 NOVEMBER 1992

SYSTEMS ANALYSIS

APPENDIX 4
INVENTORY PROGRAMME STRUCTURE AND CODE

TABLE OF CENTENTS

1. TREE DIAGRAM

```
VFI.PRG
│    PART101.DBF (database)
│    PART104.DBF (database)
│    PART103.DBF (database)
│    PDATA.DBF (database)
│    ID101.NDX (index file)
│    CODE104.NDX (index file)
│    CODE103.NDX (index file)
│    PDATA.NDX (index file)
├────SCR1(procedure in VFIPR.PRG)
├────ENTRYX(procedure in VFIEN.PRG)
│    ├────ENTRY1(proc in VFIEN.PRG)
│    │    │  PART101.DBF (database)
│    │    │  PART102.DBF (database)
│    │    │  PART103.DBF (database)
│    │    │  PART104.DBF (database)
│    │    │  PART105.DBF (database)
│    │    │  PART106.DBF (database)
│    │    ├────PART101.FMT
│    │    ├────PART102.FMT
│    │    ├────PART103.FMT
│    │    ├────PART104.FMT
│    │    ├────PART105.FMT
│    │    ├────PART106.FMT
│    │    ├────P1ENTRY(pro.VFIEN.PRG)
│    │    └────P1EDIT(proc.VFIEN.PRG)
│    └────ENTRY2(proc.VFIEN.PRG)
│         │  PART201.DBF (database)
│         │  PART202.DBF (database)
│         │  PART203.DBF (database)
│         │  PART204.DBF (database)
│         │  PART301.DBF (database)
│         ├────PART201.FMT
│         ├────PART202.FMT
│         ├────PART203.FMT
│         ├────PART204.FMT
│         ├────PART301.FMT
│         ├────P2ENTRY(proc.VFIEN.PRG)
│         │    │  PART205.DBF (database)
│         │    │  PART206.DBF (database)
│         │    │  PART207.DBF (database)
│         │    ├────PART205.FMT
│         │    ├────PART206.FMT
│         │    └────PART207.FMT
│         └────P2EDIT(proc.VFIEN.PRG)
│                 PART205.DBF (database)
│                 PART206.DBF (database)
│                 PART207.DBF (database)
└────PRNT(procedure in VFIPR.PRG)
     ├────STOCLIVE(proc.VFIPR.PRG)
     │    │  PART101.DBF (database)
     │    │  PDATA.DBF (database)
     │    │  PDATA.NDX (index file)
     │    └────H_SCR(proc.in VFIPR.PRG)
     ├────PALMLAND(proc.in VFIPR.PRG)
     │    │  PART101.DBF (database)
     │    │  PART105.DBF (database)
     │    │  PDATA.DBF (database)
     │    │  ID101.NDX (index file)
     │    │  PDATA.NDX (index file)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────VOLTREE(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PART104.DBF (database)
     │    │  PDATA.NDX (index file)
     │    │  ID101.NDX (index file)
     │    │  CODE104.NDX (index file)
     │    ├────GET_REG(proc.VFIPR.PRG)
     │    │    └────SCR1(proc.VFIPR.PRG)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────NOWOLAND(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PART106.DBF (database)
     │    │  PDATA.NDX (index file)
     │    │  ID101.NDX (index file)
     │    ├────GET_REG(proc.VFIPR.PRG)
     │    │    └────SCR1(proc.VFIPR.PRG)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────CANELAND(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PDATA.NDX (index file)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────FUELWHOS(proc.VFIPR.PRG)
     │    │  PART101.DBF (database)
     │    │  PART201.DBF (database)
     │    │  ID101.NDX (index file)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────STOCSMAL(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PART103.DBF (database)
     │    │  PDATA.NDX (index file)
     │    │  ID101.NDX (index file)
     │    │  CODE103.NDX (index file)
     │    ├────GET_REG(proc.VFIPR.PRG)
     │    │    └────SCR1(proc.VFIPR.PRG)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────STOCBAM(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PART102.DBF (database)
     │    │  PDATA.NDX (index file)
     │    │  ID101.NDX (index file)
     │    ├────GET_REG(proc.VFIPR.PRG)
     │    │    └────SCR1(proc.VFIPR.PRG)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────FUELFAR(proc.VFIPR.PRG)
     │    │  PART101.DBF (database)
     │    │  PART201.DBF (database)
     │    │  ID101.NDX (index file)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────FUELAVAI(proc.VFIPR.PRG)
     │    │  PART101.DBF (database)
     │    │  PART201.DBF (database)
     │    │  ID101.NDX (index file)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────STOCNOWO(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PART106.DBF (database)
     │    │  PDATA.NDX (index file)
     │    │  ID101.NDX (index file)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────BAMLAND(proc.VFIPR.PRG)
     │    │  PDATA.DBF (database)
     │    │  PART101.DBF (database)
     │    │  PART102.DBF (database)
     │    │  PDATA.NDX (index file)
     │    │  ID101.NDX (index file)
     │    ├────GET_REG(proc.VFIPR.PRG)
     │    │    └────SCR1(proc.VFIPR.PRG)
     │    └────H_SCR(proc.VFIPR.PRG)
     ├────GRASLAND(proc.VFIPR.PRG)
     │    │  PART101.DBF (database)
     │    └────H_SCR(proc.VFIPR.PRG)
     │
```

```
├──FUELUSE(proc.VFIPR.PRG)
│  │  PART101.DBF (database)
│  │  PART201.DBF (database)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──FUELDAYS(proc.VFIPR.PRG)
│  │  PART101.DBF (database)
│  │  PART201.DBF (database)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──FODDFAR(proc.VFIPR.PRG)
│  │  PART101.DBF (database)
│  │  PART201.DBF (database)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──FODDAVAI(proc.VFIPR.PRG)
│  │  PART101.DBF (database)
│  │  PART201.DBF (database)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──TRANSUSE(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PART205.DBF (database)
│  │  PDATA.NDX (index file)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──BUILDUSE(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PART202.DBF (database)
│  │  PDATA.NDX (index file)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──SALESPUR(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PART206.DBF (database)
│  │  PDATA.NDX (index file)
│  │  ID101.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──OCCULAND(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PDATA.NDX (index file)
│  ├───GET_REG(proc.VFIPR.PRG)
│  │  └───SCR1(proc.VFIPR.PRG)
│  └───H_SCR(proc.VFIPR.PRG)
│
├──BAMCLUMP(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PART102.DBF (database)
│  │  PDATA.NDX (index file)
│  │  ID101.NDX (index file)
│  ├───GET_REG(proc.VFIPR.PRG)
│  │  └───SCR1(proc.VFIPR.PRG)
│  └───H_SCR(proc.VFIPR.PRG)
├──STOCPALM(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PART105.DBF (database)
│  │  PDATA.NDX (index file)
│  │  ID101.NDX (index file)
│  ├───GET_REG(proc.VFIPR.PRG)
│  │  └───SCR1(proc.VFIPR.PRG)
│  └───H_SCR(proc.VFIPR.PRG)
├──SMALLAND(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART101.DBF (database)
│  │  PART1031.DBF (database)
│  │  PDATA.NDX (index file)
│  │  ID101.NDX (index file)
│  │  CODE1031.NDX (index file)
│  ├───GET_REG(proc.VFIPR.PRG)
│  │  └───SCR1(proc.VFIPR.PRG)
│  └───H_SCR(proc.VFIPR.PRG)
├──STOCMID(proc.VFIPR.PRG)
│  │  PDATA.DBF (database)
│  │  PART104.DBF (database)
│  │  PDATA.NDX (index file)
│  │  CODE104.NDX (index file)
│  └───H_SCR(proc.VFIPR.PRG)
├──POPULAND(proc.VFIPR.PRG)
│  │  PART101.DBF (database)
│  └───H_SCR(proc.VFIPR.PRG)
└──TREELAND(proc.VFIPR.PRG)
   │  PDATA.DBF (database)
   │  PART101.DBF (database)
   │  PART104.DBF (database)
   │  PDATA.NDX (index file)
   │  ID101.NDX (index file)
   │  CODE104.NDX (index file)
   ├───GET_REG(proc.VFIPR.PRG)
   │  └───SCR1(proc.VFIPR.PRG)
   └───H_SCR(proc.VFIPR.PRG)
```

2. PROGRAMME CODE

```
*:*************************************
*:    Program: VFI.PRG
*:    Calls
*:          : SCR1              (proc.in VFIPR.PRG)
*:          : ENTRYX            (proc.in VFIEN.PRG)
*:          : PRNT              (proc.in VFIPR.PRG)
*:    Uses
*:          : PART101.DBF
*:          : PART104.DBF
*:          : PART103.DBF
*:          : PDATA.DBF
*:    Indexes
*:          : ID101.NDX
*:          : CODE104.NDX
*:          : CODE103.NDX
*:          : PDATA.NDX
*:*************************************
CLEAR ALL
SET TALK OFF
SET ECHO OFF
SET STATUS OFF
SET DATE ITALIAN
SET INTENSITY ON
SET BELL OFF
SET FIXED ON
SET DELETE ON
SET ESCAPE OFF
SET PROC TO VFIPR
SET COLOR TO W/B,N/W
IF .NOT. FILE("ID101.NDX")
    USE PART101
    INDEX ON ID TO ID101
ENDIF
IF .NOT. FILE("CODE104.NDX")
    USE PART104
    INDEX ON CODE TO CODE104
ENDIF
IF .NOT. FILE("CODE103.NDX")
    USE PART103
    INDEX ON CODE TO CODE103
ENDIF
IF .NOT. FILE("PDATA.NDX")
    USE PDATA
    INDEX ON STRTA TO PDATA
ENDIF
USE
R_S1="1 - North West"
R_S2="2 - North Central"
R_S3="3 - West"
R_S4="4 - South"
R_S5="5 - South East"
R_S6="6 - North East"
T=1
```

```
TITLE1="FORESTRY MASTER PLAN"
TITLE2="VILLAGE FOREST INVENTORY
                                PROGRAMME"
TITLE5="Press any NUMBER within
                                this MENU"
TDATE=TRIM(UPPER(CMONTH
(DATE())))+" "+TRIM(LTRIM
                       (STR(DAY(DATE()))))+",

"+TRIM(LTRIM(STR(YEAR(DATE()))))
DO SCR1
DO WHILE .T.
     @ 8,30 TO 14,46 DOUBLE
     @ 10,31 SAY "———————————————"
     @ 12,31 SAY "———————————————"
     VCHOICE=0
     SET COLOR TO W/B
     @  9, 31  SAY " 1 - ENTRY    "
     @ 11, 31  SAY " 2 - PRINT    "
     @ 13, 31  SAY " 3 - EXIT     "
     @ 23,(80-LEN(TITLE5))/2 SAY TITLE5
     @22,30 SAY "Enter Choice: "
                      GET VCHOICE PICTURE
                               "9" RANGE 1,3
     READ
     DO CASE
     CASE VCHOICE=3
          SET STATUS ON
          SET TALK ON
          SET BELL ON
          CLEAR ALL
          CLEAR
          CLOSE ALL
          RETURN
     CASE VCHOICE = 1
          DO SCR1
          CLOSE PROC
          SET PROC TO VFIEN
          DO ENTRYX
          CLOSE PROC
          SET PROC TO VFIPR
     CASE VCHOICE = 2
          DO SCR1
          DO PRNT
     ENDCASE
     CLOSE DATABASES
     DO SCR1
ENDDO
RETURN
***********************************
```

```
*:**************************************
*:    Procedure file: VFIEN.PRG
*:    Procs & Fncts
*:          : ENTRYX
*:          : ENTRY1
*:          : P1ENTRY
*:          : P1EDIT
*:          : ENTRY2
*:          : P2ENTRY
*:          : P2EDIT
*:**************************************
*!   Procedure: ENTRYX
*!   Called by: VFI.PRG
*!   Calls
*!          : ENTRY1(proc. in VFIEN.PRG)
*!          : ENTRY2(proc. in VFIEN.PRG)
*!**************************************PROC
ENTRYX
@5,0 CLEAR TO 20,79
E_S=0
@10,28 SAY "1 - PART I"
@12,28 SAY "2 - PART II & III"
@22,28 SAY "Enter Choice:" GET E_S PICTURE "9"
RANGE 0,2
READ
IF E_S#0
    DO CASE
    CASE E_S=1
        DO ENTRY1
    CASE E_S=2
        DO ENTRY2
    ENDCASE
ENDIF
RETURN
*!**************************************
*!   Procedure: ENTRY1
*!   Called by
*!          : ENTRY(proc. in VFIPR.PRG)
*!          : ENTRYX (proc. in VFIEN.PRG)
*!   Calls
*!          : PART101.FMT
*!          : PART102.FMT
*!          : PART103.FMT
*!          : PART104.FMT
*!          : PART105.FMT
*!          : PART106.FMT
*!          : P1ENTRY (proc. in VFIEN.PRG)
*!          : P1EDIT (proc. in VFIEN.PRG)
*!   Uses
*!          : PART101.DBF
*!          : PART102.DBF
*!          : PART103.DBF
*!          : PART104.DBF
*!          : PART105.DBF
*!          : PART106.DBF
*!   Formats
*!          : PART101.FMT
*!          : PART102.FMT
*!          : PART103.FMT
*!          : PART104.FMT
*!          : PART105.FMT
*!          : PART106.FMT
*!**************************************
PROC ENTRY1
SELECT A
USE PART101
SET FORMAT TO PART101
SELECT B
USE PART102
SET FORMAT TO PART102
SELECT C
```

```
USE PART103
SET FORMAT TO PART103
SELECT D
USE PART104
SET FORMAT TO PART104
SELECT E
USE PART105
SET FORMAT TO PART105
SELECT F
USE PART106
SET FORMAT TO PART106
DO WHILE .T.
     CLEAR
     TEXT
          *********** OPENING MENU
                                        ************
                 1 >   DATA ENTRY

                 2 >   VIEW/EDIT

                 Q >   QUIT

           ************
                            ***************************
     ENDTEXT
     ?
     WAIT '          MAKE YOUR
                               CHOICE > ' TO CH
     DO CASE
     CASE CH='1'
         DO P1ENTRY
     CASE CH='2'
         DO P1EDIT
     CASE UPPER(CH)='Q'
         CLEAR
         EXIT
     ENDCASE
ENDDO
CLOSE DATABASES
RETURN
*!**************************************
*!   Procedure: P1ENTRY
*!   Called by: ENTRY1 (proc.in VFIEN.PRG)
*!**************************************
PROCEDURE P1ENTRY
DO WHILE .T.
     CLEAR
     WAIT '   DO YOU WANT TO ENTER
                  DATA?  <Y/N>: ' TO CHOICE
     CLEAR
     IF UPPER(CHOICE)='N'
          EXIT
     ENDIF
     SELECT A
     APPEND
     READ
     STORE ID TO SNO
     CLEAR
     WAIT 'IS THERE ANY ENTRY FOR
                 BAMBOO?  <Y/N>: ' TO CHECK
     CLEAR
     IF UPPER(CHECK)='Y'
          SELECT B
          APPEND
          READ
          REPLACE ID WITH SNO
          CLEAR
     ENDIF
     WAIT 'IS THERE ENTRY FOR TREE
                   OF SMALL GIRTH?  <Y/N>: '
                                   TO CHECK
```

```
    CLEAR
    IF UPPER(CHECK)='Y'
         SELECT C
         GO BOTTOM
         NUM=RECNO()
         APPEND
         READ
         IF NUM=1
              GO 1
              REPLACE ID WITH SNO
              GO BOTTOM
         ENDIF
         NOM=RECNO()
         IF NUM<>NOM
              GO (NUM+1)
              REPLACE REST ID WITH SNO
         ENDIF
         CLEAR
    ENDIF
    SELECT D
    GO BOTTOM
    NUM=RECNO()
    APPEND
    READ
    IF NUM=1
         GO 1
         REPLACE ID WITH SNO
         GO BOTTOM
    ENDIF
    NOM=RECNO()
    IF NUM<>NOM
         GO (NUM+1)
         REPLACE REST ID WITH SNO
    ENDIF
    CLEAR
    WAIT 'IS THERE  ENTRY FOR PALM
                   TREE?  <Y/N>: ' TO CHECK
    CLEAR
    IF UPPER(CHECK)='Y'
         SELECT E
         APPEND
         READ
         REPLACE ID WITH SNO
         CLEAR
    ENDIF
    WAIT 'IS THERE ENTRY FOR TREE
                     OF NO VALUE?  <Y/N>: '
                                        TO CHECK
    CLEAR
    IF UPPER(CHECK)='Y'
         SELECT F
         APPEND
         READ
         REPLACE ID WITH SNO
         CLEAR
    ENDIF
ENDDO
RETURN
*!***************************************
*!   Procedure: P1EDIT
*!   Called by: ENTRY1 (proc. VFIEN.PRG)
*!***************************************
Proc. P1EDIT
DO WHILE .T.
    CLEAR
    TEXT
         ********************
                                   **************

          1 >   GENERAL INFORMATION

          2 >  BAMBOOS

          3 >  THIN TREES

          4 >  LARGE TREES

          5 >  PALM TREES

          6 >  NO WOOD VALUE

          Q >  QUIT

         **************
                        ***********************
    ENDTEXT
    ?
    WAIT '      MAKE YOUR CHOICE >'
                                   TO CHECK
    CLEAR
    IF CHECK='1'
         SELECT A
         BROWSE
    ENDIF
    IF CHECK='2'
         SELECT B
         BROWSE
    ENDIF
    IF CHECK='3'
         SELECT C
         BROWSE
    ENDIF
    IF CHECK='4'
         SELECT D
         BROWSE
    ENDIF
    IF CHECK='5'
         SELECT E
         BROWSE
    ENDIF
    IF CHECK='6'
         SELECT F
         BROWSE
    ENDIF
    IF UPPER(CHECK)='Q'
         EXIT
         LOOP
    ENDIF
ENDDO
CLEAR
RETURN
*!***************************************
*!     Procedure: ENTRY2
*!     Called by
*!          : ENTRY (proc. in VFIPR.PRG)
*!          : ENTRYX (proc. in VFIEN.PRG)
*!     Calls
*!          : PART201.FMT
*!          : PART202.FMT
*!          : PART203.FMT
*!          : PART204.FMT
*!          : PART301.FMT
*!          : P2ENTRY (proc.in VFIEN.PRG)
*!          : P2EDIT (proc.in VFIEN.PRG)
*!     Uses
*!          : PART201.DBF
*!          : PART202.DBF
*!          : PART203.DBF
*!          : PART204.DBF
*!          : PART301.DBF
*!     Formats
*!          : PART201.FMT
```

```
          : PART202.FMT
*!        : PART203.FMT
*!        : PART204.FMT
*!        : PART301.FMT
*!
*!*************************************
PROC ENTRY2
SELECT A
USE PART201
SET FORMAT TO PART201
SELECT B
USE PART202
SET FORMAT TO PART202
SELECT C
USE PART203
SET FORMAT TO PART203
SELECT D
USE PART204
SET FORMAT TO PART204
SELECT F
USE PART301
SET FORMAT TO PART301
DO WHILE .T.
    CLEAR
    TEXT
         ************ OPENING MENU
                                  *************

              1 >   DATA ENTRY

              2 >   VIEW/EDIT

              Q >   QUIT

         *******************
                                  ********************
    ENDTEXT
    ?
    WAIT '          MAKE YOUR
                            CHOICE > ' TO CH
    DO CASE
    CASE CH='1'
        DO P2ENTRY
    CASE CH='2'
        DO P2EDIT
    CASE UPPER(CH)='Q'
        CLEAR
        EXIT
    ENDCASE
ENDDO
RETURN
*!*************************************
*!    Procedure: P2ENTRY
*!    Called by: ENTRY2 (proc.in VFIEN.PRG)
*!    Calls
*!        : PART205.FMT
*!        : PART206.FMT
*!        : PART207.FMT
*!    Uses
*!        : PART205.DBF
*!        : PART206.DBF
*!        : PART207.DBF
*!    Formats
*!        : PART205.FMT
*!        : PART206.FMT
*!        : PART207.FMT
*!*************************************
Proc. P2ENTRY
DO WHILE .T.
    CLEAR
    WAIT '   DO YOU WANT TO ENTER
                DATA?  <Y/N>: ' TO CHOICE
```

```
CLEAR
IF UPPER(CHOICE)='N'
    EXIT
ENDIF
SELECT A
APPEND
READ
STORE ID TO SNO
CLEAR
SELECT B
APPEND
READ
REPLACE ID WITH SNO
CLEAR
SELECT C
APPEND
READ
REPLACE ID WITH SNO
CLEAR
SELECT D
APPEND
READ
REPLACE REST ID WITH SNO
CLEAR
SELECT E
WAIT 'Is there any entry for
              Transportation <y/n>? ' TO CK
IF UPPER(CK)='Y'
    CLEAR
    USE PART205
    SET FORMAT TO PART205
    APPEND
    READ
    REPLACE ID WITH SNO
ENDIF
CLEAR
WAIT 'Is there any entry for Sales
              & Purchase <y/n>? ' TO CK
IF UPPER(CK)='Y'
    CLEAR
    USE PART206
    SET FORMAT TO PART206
    APPEND
    READ
    REPLACE ID WITH SNO
ENDIF
CLEAR
WAIT 'Is there any entry for
                   Harvest <y/n>? ' TO CK
IF UPPER(CK)='Y'
    CLEAR
    USE PART207
    SET FORMAT TO PART207
    GO BOTTOM
    NUM = RECNO()
    APPEND
    READ
    IF NUM=1
        GO 1
        REPLACE ID WITH SNO
        GO BOTTOM
    ENDIF
    NOM = RECNO()
    IF NUM<>NOM
        GO (NUM+1)
        REPLACE REST ID WITH SNO
    ENDIF
ENDIF
CLEAR
SELECT F
APPEND
```

```
    READ
    REPLACE ID WITH SNO
    CLEAR
ENDDO
RETURN

*!*************************************
*!   Procedure: P2EDIT
*!   Called by: ENTRY2 (proc.in VFIEN.PRG)
*!   Uses
*!          : PART205.DBF
*!          : PART206.DBF
*!          : PART207.DBF
*!*************************************
Proc. P2EDIT
DO WHILE .T.
    CLEAR
    TEXT
         ****************
                              ********************

     1 >  FUEL AND FODDER

     2 >  BUILDING MATERIALS

     3 >  FURNITURE

     4 >  AGRICULTURAL IMPLEMENTS

     5 >  TRANSPORTATION

     6 >  SALES & PURCHASE

     7 >  HARVEST

     8 >  ATTITUDE

     Q >  QUIT   -

         *****************
                              ********************
    ENDTEXT
    ?
    WAIT '     MAKE YOUR
                          CHOICE > ' TO CHECK
    CLEAR
    IF CHECK='1'
         SELECT A
         BROWSE
    ENDIF
    IF CHECK='2'
         SELECT B
         BROWSE
    ENDIF
    IF CHECK='3'
         SELECT C
         BROWSE
    ENDIF
    IF CHECK='4'
         SELECT D
         BROWSE
    ENDIF
    IF CHECK='5'
         SELECT E
         USE PART205
         BROWSE
    ENDIF
    IF CHECK='6'
         SELECT E
         USE PART206
         BROWSE
    ENDIF
    IF CHECK='7'
         SELECT E
         USE PART207
         BROWSE
    ENDIF
    IF CHECK='8'
         SELECT F
         BROWSE
    ENDIF
    IF UPPER(CHECK)='Q'
         EXIT
         LOOP
    ENDIF
ENDDO
CLEAR
RETURN
*:*************************************
*:   Procedure file: VFIPR.PRG
*:   Procedures & Functions
*:          : PROCS
*:          : ENTRY
*:          : SCR1
*:          : SCR2
*:          : PRNT
*:          : GET_REG
*:          : STOCLIVE
*:          : H_SCR
*:          : PALMLAND
*:          : VOLTREE
*:          : NOWOLAND
*:          : CANELAND
*:          : FUELWHOS
*:          : STOCSMAL
*:          : STOCBAM
*:          : FUELFAR
*:          : FUELAVAI
*:          : STOCNOWO
*:          : BAMLAND
*:          : GRASLAND
*:          : FUELUSE
*:          : FUELDAYS
*:          : FODDFAR
*:          : FODDAVAI
*:          : TRANSUSE
*:          : BUILDUSE
*:          : SALESPUR
*:          : OCCULAND
*:          : BAMCLUMP
*:          : STOCPALM
*:          : SMALLAND
*:          : STOCMID
*:          : POPULAND
*:          : TREELAND
*:*************************************
*!   Procedure: PROCS
*!   Calls
*!          : ENTRY (proc.in VFIPR.PRG)
*!          : SCR1 (proc.in VFIPR.PRG)
*!          : PRNT (proc.in VFIPR.PRG)
*!          : SCR2 (proc.in VFIPR.PRG)
*!*************************************
PROC PROCS
DO CASE
CASE VCHOICE = 1
     DO ENTRY
CASE VCHOICE = 2
     DO SCR1
     DO PRNT
ENDCASE
CLOSE DATABASES
```

```
DO SCR1
DO SCR2
RETURN
*!**************************************
*!    Procedure: ENTRY
*!    Called by: PROCS (proc.in VFIPR.PRG)
*!    Calls
*!          : SCR1 (proc.in VFIPR.PRG)
*!          : ENTRY1 (proc.in VFIEN.PRG)
*!          : ENTRY2 (proc.in VFIEN.PRG)
*!**************************************
PROC ENTRY
DO SCR1
@5,0 CLEAR TO 20,79
E_S=0
@10,28 SAY "1 - PART I"
@12,28 SAY "2 - PART II & III"
@22,28 SAY "Enter Choice:" GET E_S
                              PICTURE "9" RANGE 0,2
READ
DO CASE
CASE E_S=1
     DO ENTRY1
CASE E_S=2
     DO ENTRY2
ENDCASE
RETURN
*!**************************************
*!    Procedure: SCR1
*!    Called by
*!          : VFI.PRG
*!          : PROCS (proc.in VFIPR.PRG)
*!          : ENTRY (proc.in VFIPR.PRG)
*!          : GET_REG (proc.in VFIPR.PRG)
*!**************************************
PROC SCR1
CLEAR
@  2, (80-LEN(TITLE1))/2 SAY TITLE1
@  3, (80-LEN(TITLE2))/2  SAY TITLE2
@  0, 72 SAY TIME()
@  0, 0  SAY TDATE
@  1, 0 SAY REPL(CHR(177),80)
@  4, 0 SAY REPL(CHR(177),80)
@ 21, 0  SAY REPL(CHR(177),80)
@ 24, 0  SAY REPL(CHR(177),80)
X=1
DO WHILE X<3
     @X+1,0 SAY CHR(177)+CHR(177)
     @X+1,78 SAY CHR(177)+CHR(177)
     @X+21,0 SAY CHR(177)+CHR(177)
     @X+21,78 SAY CHR(177)+CHR(177)
     X=X+1
ENDDO
SET COLOR TO N/W
@5,0 CLEAR TO 20,79
SET COLOR TO W/B,N/W
RETURN
*!**************************************
*!    Procedure: SCR2
*!    Called by: PROCS (proc.in VFIPR.PRG)
*!**************************************
PROC SCR2
@ 8,30 TO 14,46 DOUBLE
@ 10,31 SAY "________________"
@ 12,31 SAY "________________"
@  9, 31   SAY " 1 - ENTRY      "
@ 11, 31   SAY " 4 - PRINT      "
@ 13, 31   SAY " 4 - EXIT       "
@ 22,(80-LEN(TITLE4))/2 SAY TITLE4
@ 23,(80-LEN(TITLE5))/2 SAY TITLE5
RETURN
```

```
*!**************************************
*!    Procedure: PRNT
*!    Called by
*!          : VFI.PRG
*!          : PROCS (proc.in VFIPR.PRG)
*!    Calls
*!          : STOCLIVE (proc.in VFIPR.PRG)
*!          : PALMLAND (proc.in VFIPR.PRG)
*!          : VOLTREE (proc.in VFIPR.PRG)
*!          : NOWOLAND (proc.in VFIPR.PRG)
*!          : CANELAND (proc.in VFIPR.PRG)
*!          : FUELWHOS (proc.in VFIPR.PRG)
*!          : STOCSMAL (proc.in VFIPR.PRG)
*!          : STOCBAM (proc.in VFIPR.PRG)
*!          : FUELFAR (proc.in VFIPR.PRG)
*!          : FUELAVAI (proc.in VFIPR.PRG)
*!          : STOCNOWO (proc.in VFIPR.PRG)
*!          : BAMLAND (proc.in VFIPR.PRG)
*!          : GRASLAND (proc.in VFIPR.PRG)
*!          : FUELUSE (proc.in VFIPR.PRG)
*!          : FUELDAYS (proc.in VFIPR.PRG)
*!          : FODDFAR (proc.in VFIPR.PRG)
*!          : FODDAVAI (proc.in VFIPR.PRG)
*!          : TRANSUSE (proc.in VFIPR.PRG)
*!          : BUILDUSE (proc.in VFIPR.PRG)
*!          : SALESPUR (proc.in VFIPR.PRG)
*!          : OCCULAND (proc.in VFIPR.PRG)
*!          : BAMCLUMP (proc.in VFIPR.PRG)
*!          : STOCPALM (proc.in VFIPR.PRG)
*!          : SMALLAND (proc.in VFIPR.PRG)
*!          : STOCMID (proc.in VFIPR.PRG)
*!          : POPULAND (proc.in VFIPR.PRG)
*!          : TREELAND (proc.in VFIPR.PRG)
*!**************************************
PROC PRNT
SET COLOR TO N/W
@6,1 SAY " 1: LIVESTOCK NUMBER"
@7,1 SAY " 2: STOCK OF PALM TREES:
                                   ALL STRATA"
@8,1 SAY " 3: STOCK VOLUME & STAND
                                        TABLE"
@9,1 SAY " 4: STOCK OF TREES OF NO
                                  WOOD VALUE"
@10,1 SAY " 5: CANES: NO OF CULMS"
@11,1 SAY " 6: FUEL COLLECTORS"
@12,1 SAY ' 7: TREE STOCK:  DIA<=8"'+
                                  " & HT.>5'"
@13,1 SAY " 8: BAMBOO RESOURCES"
@14,1 SAY " 9: DISTANCE TO GO TO
                                COLLECT FUEL"
@15,1 SAY "10: RANKING OF
                        AVAILABILITY OF FUEL"
@16,1 SAY "11: TREE STOCK-NO WOOD
                                  VALUE: ALL"
@17,1 SAY "12: BAMBOO
                   RESOURCES-LANDOWNERSHIP"
@18,1 SAY "13: THATCH GRASS AREA"
@19,1 SAY "14: FUEL CONSUMPTION:
                                  ALL STRATA"
@6,38 SAY "15: DAYS OF WEEK IN FUEL
                                  COLLECTION"
@7,38 SAY "16: DISTANCE TO GO TO
                             COLLECT FODDER"
@8,38 SAY "17: RANKING OF
                     AVAILABILITY OF FODDER"
@9,38 SAY "18: WOOD/BAMBOO USE
                             -TRANSPORTATION"
@10,38 SAY "19: WOOD/BAMBOO USE
                        -BUILDING & FENCING"
@11,38 SAY "20: SALES & PURCHASE OF
                           TIMBER & BAMBOO"
```

```
@12,38 SAY "21: POPULATION/
               OCCUPATION/LANDOWNERSHIP"
@13,38 SAY "22: BAMBOO RESOURCES -
                                CLUMP"
@14,38 SAY "23: STOCK OF PALM TREES"
@15,38 SAY "24: TREE STOCK-
              LANDOWNER: DIA<=8"'+", HT>5'"
@16,38 SAY "25: STOCK OF TREES: 4"
                              < DIA <8"'
@17,38 SAY "26: POPULATION
                          DISTRIBUTION"
@18,38 SAY "27: STOCK VOL. BY
                           LANDHOLDING"
SET COLOR TO W/B
P_CH=0
P_P='N'
@22,24 SAY "Enter Number to Print-------->"
             GET P_CH PICTURE "99" RANGE 0,27
@23,24 SAY "Send Output to Printer?(y/n)->"
                        GET P_P PICTURE "Y"
READ
DO CASE
CASE  P_CH=1
    DO STOCLIVE
CASE  P_CH=2
    DO PALMLAND
CASE  P_CH=3
    DO VOLTREE
CASE  P_CH=4
    DO NOWOLAND
CASE  P_CH=5
    DO CANELAND
CASE  P_CH=6
    DO FUELWHOS
CASE  P_CH=7
    DO STOCSMAL
CASE  P_CH=8
    DO STOCBAM
CASE  P_CH=9
    DO FUELFAR
CASE  P_CH=10
    DO FUELAVAI
CASE  P_CH=11
    DO STOCNOWO
CASE  P_CH=12
    DO BAMLAND
CASE  P_CH=13
    DO GRASLAND
CASE  P_CH=14
    DO FUELUSE
CASE  P_CH=15
    DO FUELDAYS
CASE  P_CH=16
    DO FODDFAR
CASE  P_CH=17
    DO FODDAVAI
CASE  P_CH=18
    DO TRANSUSE
CASE  P_CH=19
    DO BUILDUSE
CASE  P_CH=20
    DO SALESPUR
CASE  P_CH=21
    DO OCCULAND
CASE  P_CH=22
    DO BAMCLUMP
CASE  P_CH=23
    DO STOCPALM
CASE  P_CH=24
    DO SMALLAND
CASE  P_CH=25
    DO STOCMID
CASE  P_CH=26
    DO POPULAND
CASE  P_CH=27
    DO TREELAND
OTHERWISE
    RETURN
ENDCASE
CLOSE DATABASES
RETURN
*!***************************************
*!    Procedure: GET_REG
*!    Called by
*!          : VOLTREE (proc.in VFIPR.PRG)
*!          : NOWOLAND (proc.in VFIPR.PRG)
*!          : STOCSMAL (proc.in VFIPR.PRG)
*!          : STOCBAM (proc.in VFIPR.PRG)
*!          : BAMLAND (proc.in VFIPR.PRG)
*!          : OCCULAND (proc.in VFIPR.PRG)
*!          : BAMCLUMP (proc.in VFIPR.PRG)
*!          : STOCPALM (proc.in VFIPR.PRG)
*!          : SMALLAND (proc.in VFIPR.PRG)
*!          : TREELAND (proc.in VFIPR.PRG)
*!    Calls: SCR1 (proc.in VFIPR.PRG)
*!***************************************
PROC GET_REG
DO SCR1
@5,0 CLEAR TO 20,79
@8,28 SAY "   REGIONS"
@10,28 SAY "1 - NORTH WEST"
@11,28 SAY "2 - NORTH CENTRAL"
@12,28 SAY "3 - WEST"
@13,28 SAY "4 - SOUTH"
@14,28 SAY "5 - SOUTH EAST"
@15,28 SAY "6 - NORTH EAST"
@22,28 SAY "Enter Choice:" GET S_S PICTURE "9"
RANGE 0,6
READ
RETURN
*!***************************************
*!    Procedure: STOCLIVE
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!          : PART101.DBF
*!          : PDATA.DBF
*!    Indexes: PDATA.NDX
*!***************************************
PROC STOCLIVE
SELE A
USE PART101
STORE 0 TO CA1,BU1,GT1,SH1,OT1
STORE 0 TO CA2,BU2,GT2,SH2,OT2
STORE 0 TO CA3,BU3,GT3,SH3,OT3
STORE 0 TO CA4,BU4,GT4,SH4,OT4
STORE 0 TO CA5,BU5,GT5,SH5,OT5
STORE 0 TO CA6,BU6,GT6,SH6,OT6
DO WHILE .NOT. EOF()
     S_S=SUBSTR(SAMPLE,1,1)
     CA&S_S=CA&S_S+CATTLE
     BU&S_S=BU&S_S+BUFFALO
     GT&S_S=GT&S_S+GOAT
     SH&S_S=SH&S_S+SHEEP
     OT&S_S=OT&S_S+OTH_LIVE
     SKIP
ENDDO
USE PDATA INDEX PDATA
CLEAR
SP=SPACE(4)
IF P_P='Y'
     SET DEVICE TO PRINT
```

```
ENDIF
@1,5 SAY 'LIVESTOCK'
@2,5 SAY REPL(CHR(196),66)
@3,5 SAY '              Cattle    Buffalo
                              Goat      Sheep      Others'
@4,5 SAY '             --------  -------  -------
                                        -------   -------'
X=1
P=6
DO WHILE X<7
     XX=STR(X,1,0)
     SEEK XX
     @P,5 SAY "Region: "+R_S&XX
     @P+2,5 SAY 'Number            '+
                    TRAN(CA&XX,'9999999')+SP+
                    TRAN(BU&XX,'9999999')+SP+
                    TRAN(GT&XX,'9999999')+SP+
                    TRAN(SH&XX,'9999999')+SP+
                         TRAN(OT&XX,'9999999')
     @P+3,5 SAY 'Per Capita      '+
               TRAN(CA&XX/S_PT,'9999.99')+SP+
               TRAN(BU&XX/S_PT,'9999.99')+SP+
               TRAN(GT&XX/S_PT,'9999.99')+SP+
               TRAN(SH&XX/S_PT,'9999.99')+SP+
                    TRAN(OT&XX/S_PT,'9999.99')
     @P+4,5 SAY REPL(CHR(196),66)
     X=X+1
     P=P+5
     IF P_P='N' .AND. P=21
          DO H_SCR
          @6,0 CLEAR TO 24,79
          P=6
     ENDIF
ENDDO
IF P_P='Y'
     @0,0 SAY " "
     SET DEVICE TO SCREEN
ENDIF
RETURN

*!*****************************************
*!    Procedure: H_SCR
*!    Called by
*!          : STOCLIVE (proc.in VFIPR.PRG)
*!          : PALMLAND (proc.in VFIPR.PRG)
*!          : VOLTREE (proc.in VFIPR.PRG)
*!          : NOWOLAND (proc.in VFIPR.PRG)
*!          : CANELAND (proc.in VFIPR.PRG)
*!          : FUELWHOS (proc.in VFIPR.PRG)
*!          : STOCSMAL (proc.in VFIPR.PRG)
*!          : STOCBAM (proc.in VFIPR.PRG)
*!          : FUELFAR (proc.in VFIPR.PRG)
*!          : FUELAVAI (proc.in VFIPR.PRG)
*!          : STOCNOWO (proc.in VFIPR.PRG)
*!          : BAMLAND (proc.in VFIPR.PRG)
*!          : GRASLAND (proc.in VFIPR.PRG)
*!          : FUELUSE (proc.in VFIPR.PRG)
*!          : FUELDAYS (proc.in VFIPR.PRG)
*!          : FODDFAR (proc.in VFIPR.PRG)
*!          : FODDAVAI (proc.in VFIPR.PRG)
*!          : TRANSUSE (proc.in VFIPR.PRG)
*!          : BUILDUSE (proc.in VFIPR.PRG)
*!          : SALESPUR (proc.in VFIPR.PRG)
*!          : OCCULAND (proc.in VFIPR.PRG)
*!          : BAMCLUMP (proc.in VFIPR.PRG)
*!          : STOCPALM (proc.in VFIPR.PRG)
*!          : SMALLAND (proc.in VFIPR.PRG)
*!          : STOCMID (proc.in VFIPR.PRG)
*!          : POPULAND (proc.in VFIPR.PRG)
*!          : TREELAND (proc.in VFIPR.PRG)
*!*****************************************
```

```
PROC H_SCR
@24,5 SAY "Press any key to continue....."
KEY=0
DO WHILE KEY=0
     KEY=INKEY()
ENDDO
RETURN
*!*************************************
*!    Procedure: PALMLAND
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!          : PART101.DBF
*!          : PART105.DBF
*!          : PDATA.DBF
*!    Indexes
*!          : ID101.NDX
*!          : PDATA.NDX
*!*************************************
PROC PALMLAND
SELECT A
USE PART101 INDEX ID101
SELECT B
USE PART105
SET RELATION TO ID INTO A
STORE 0 TO T1,T2,T3,T4,N1,N2,N3,
                    N4,K1,K2,K3,K4,S1,S2,S3,S4
DO WHILE .NOT. EOF()
     L=A->LAND
     T&L=T&L+TAL1+TAL2+TAL3
     N&L=N&L+NAR1+NAR2+NAR3
     K&L=K&L+KHE1+KHE2+KHE3
     S&L=S&L+SUP1+SUP2+SUP3
     SKIP
ENDDO
USE PDATA INDEX PDATA
SEEK 'T'
CLEAR
SP=SPACE(3)
IF P_P='Y'
     SET DEVICE TO PRINT
ENDIF
@2,2 SAY ' Stock of Palm trees in terms of
                         land ownership - All Strata'
@3,2 SAY REPLICATE(CHR(196),70)
@4,2 SAY '   Land          Tal        Narkel
                                   Khejur       Supari'
@5,2 SAY '   Acres       no.   pc    no.   pc
                                   no.   pc    no.   pc'
@6,2 SAY '-----------  ----  ----  ----  ----
                                ----  ----  ----  ----'

XX=1
P=7
STR1='Less than 0.50'
STR2='0.50 to   <2.50'
STR3='2.50 to   <7.50'
STR4='7.50 and above'
DO WHILE XX<5
     X=STR(XX,1,0)
     @P,2 SAY STR&X+SP+TRAN(T&X,'99999')+
               ' '+TRAN(T&X/S_P&X,'9.99')+SP+
                        TRAN(N&X,'99999')+' '+
                 TRAN(N&X/S_P&X,'9.99')+SP+
                        TRAN(K&X,'99999')+' '+
                 TRAN(K&X/S_P&X,'9.99')+SP+
                        TRAN(S&X,'99999')+' '+
                        TRAN(S&X/S_P&X,'9.99')

     P=P+2
     XX=XX+1
ENDDO
T=T1+T2+T3+T4
```

```
N=N1+N2+N3+N4
K=K1+K2+K3+K4
S=S1+S2+S3+S4
@14,2 SAY REPLICATE('-',70)
@15,18 SAY T PICTURE '999999'
@15,32 SAY N PICTURE '999999'
@15,46 SAY K PICTURE '999999'
@15,60 SAY S PICTURE '999999'
@16,2 SAY REPLICATE(CHR(196),70)
@18,2 SAY 'no - number'
@19,2 SAY 'pc - per capita'
IF P_P='Y'
    @0,0 SAY " "
    SET DEVICE TO SCREEN
ELSE
    DO H_SCR
ENDIF
RETURN
*!**************************************
*!   Procedure: VOLTREE
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls
*!         : GET_REG (proc.in VFIPR.PRG)
*!         : H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!         : PDATA.DBF
*!         : PART101.DBF
*!         : PART104.DBF
*!   Indexes
*!         : PDATA.NDX
*!         : ID101.NDX
*!         : CODE104.NDX
*!**************************************
PROC VOLTREE
S_S=0
DO GET_REG
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
SP=S_PT
TP=T_POP
USE PART101 INDEX ID101
SELECT B
USE PART104 INDEX CODE104
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP
STORE 0 TO TS, TVF, TVS
LN=6
CLEAR
IF P_P='Y'
    SET DEVICE TO PRINT
ENDIF

@1,2 SAY '              Stock Volume & Stand
                              Table - Stratum'+SST
@2,3 SAY REPLICATE(CHR(196),71)
@3,2 SAY ' Species   Stems    Total vol
                 Per Capita   Sawlog vol   Fire wood'
@4,2 SAY '        (000)    (000 cm)
                  (cu m)     (000 cm)     (000 cm)'
@5,3 SAY REPLICATE('-',71)
DO WHILE CODE='01'
    DBH=GIRTH/3.14159
    IF BOLE_HT>0
        DBOT=((DBH*BOLE_HT*12)-
                        (8*54))/(BOLE_HT*12-54)
        DMID=(DBOT+8)/2
    ELSE
        DBOT=0
        DMID=0
    ENDIF
    BT2=.54+(.039*DMID)
    VF=(-11.0739+(.2576*DBH^2))
    RUB=(DMID-BT2)/2
    VS=3.14159*(RUB/12)^2*BOLE_HT
    TS=TS+1
    TVF=TVF+VF
    TVS=TVS+VS
    SKIP
ENDDO
TVF=TVF*.02832
TVS=TVS*.02832
FW=TVF-TVS
PC=TVF/SP
TS=(TS/SP)*TP/1000
TVF=PC*TP/1000
TVS=(TVS/SP)*TP/1000
FW=(FW/SP)*TP/1000
@LN,4 SAY 'Mango'
@LN,13 SAY TS PICTURE '999999'
@LN,26 SAY TVF PICTURE '99999'
@LN,39 SAY PC PICTURE '9.999'
@LN,54 SAY TVS PICTURE '99999'
@LN,67 SAY FW PICTURE '99999'
T1=TS
T2=TVF
T3=TVS
T4=FW
T5=PC
STORE 0 TO TS, TVF, TVS
LN=LN+1
DO WHILE CODE='02'
    DBH=GIRTH/3.14159
    IF BOLE_HT>0
        DBOT=((DBH*BOLE_HT*12)-
                        (8*54))/(BOLE_HT*12-54)
        DMID=(DBOT+8)/2
    ELSE
        DBOT=0
        DMID=0
    ENDIF
    BT2=.79+(.025*DMID)
    VF=(-6.9127+(.1824*DBH^2))
    RUB=(DMID-BT2)/2
    VS=3.14159*(RUB/12)^2*BOLE_HT
    TS=TS+1
    TVF=TVF+VF
    TVS=TVS+VS
    SKIP
ENDDO
TVF=TVF*.02832
TVS=TVS*.02832
FW=TVF-TVS
PC=TVF/SP
TS=(TS/SP)*TP/1000
TVF=PC*TP/1000
TVS=(TVS/SP)*TP/1000
FW=(FW/SP)*TP/1000
@LN,4 SAY 'Jack'
@LN,13 SAY TS PICTURE '999999'
@LN,26 SAY TVF PICTURE '99999'
@LN,39 SAY PC PICTURE '9.999'
@LN,54 SAY TVS PICTURE '99999'
@LN,67 SAY FW PICTURE '99999'
T1=T1+TS
T2=T2+TVF
T3=T3+TVS
```

```
T4=T4+FW
T5=T5+PC
DO WHILE .NOT. EOF()
   CD=CODE
   STORE 0 TO TS, TVF, TVS
   LN=LN+1
   DO WHILE CODE=CD
      DBH=GIRTH/3.14159
      IF BOLE_HT>0
         DBOT=((DBH*BOLE_HT*12)-
                  (8*54))/(BOLE_HT*12-54)
         DMID=(DBOT+8)/2
      ELSE
         DBOT=0
         DMID=0
      ENDIF
      BT2=.48+(.042*DMID)
      VF=(-2.4068+(.1801*DBH^2))
      RUB=(DMID-BT2)/2
      VS=3.14159*(RUB/12)^2*BOLE_HT
      TS=TS+1
      TVF=TVF+VF
      TVS=TVS+VS
      SKIP
   ENDDO
   TVF=TVF*.02832
   TVS=TVS*.02832
   FW=TVF-TVS
   PC=TVF/SP
   IF CD='03'
      X='Rain'
   ENDIF
   IF CD='04'
      X='Simul'
   ENDIF
   IF CD='05'
      X='Bat'
   ENDIF
   IF CD='06'
      X='Madar'
   ENDIF
   IF CD='07'
      X='Koroi'
   ENDIF
   IF CD='08'
      X='Chakua'
   ENDIF
   IF CD='09'
      X='Jam'
   ENDIF
   IF CD='10'
      X='Jiul'
   ENDIF
   IF CD='11'
      X='Gab'
   ENDIF
   IF CD='12'
      X='Tetul'
   ENDIF
   IF CD='13'
      X='Bel'
   ENDIF
   IF CD='14'
      X='Pitali'
   ENDIF
   IF CD='15'
      X='Chaatim'
   ENDIF
   IF CD='16'
      X='Kadam'
   ENDIF
   IF CD='17'
      X='Debdaru'
   ENDIF
   IF CD='18'
      X='Jarul'
   ENDIF
   IF CD='19'
      X='Sal'
   ENDIF
   IF CD='20'
      X='Segun'
   ENDIF
   IF CD='21'
      X='Garjan'
   ENDIF
   IF CD='22'
      X='Palash'
   ENDIF
   IF CD='23'
      X='Lichu'
   ENDIF
   IF CD='24'
      X='Others'
   ENDIF
   TS=(TS/SP)*TP/1000
   TVF=PC*TP/1000
   TVS=(TVS/SP)*TP/1000
   FW=(FW/SP)*TP/1000
   @LN,4 SAY X
   @LN,13 SAY TS PICTURE '999999'
   @LN,26 SAY TVF PICTURE '99999'
   @LN,39 SAY PC PICTURE '9.999'
   @LN,54 SAY TVS PICTURE '99999'
   @LN,67 SAY FW PICTURE '99999'
   T1=T1+TS
   T2=T2+TVF
   T3=T3+TVS
   T4=T4+FW
   T5=T5+PC
   IF P_P='N' .AND. LN=20 .AND.
                                            .NOT. EOF()
      DO H_SCR
      @6,0 CLEAR TO 24,79
      LN=5
   ENDIF
ENDDO
LN=LN+1
@LN,3 SAY REPLICATE(CHR(196),71)
@LN+1,13 SAY T1 PICTURE '999999'
@LN+1,26 SAY T2 PICTURE '99999'
@LN+1,39 SAY T5 PICTURE '9.999'
@LN+1,54 SAY T3 PICTURE '99999'
@LN+1,67 SAY T4 PICTURE '99999'
@LN+2,3 SAY REPLICATE(CHR(196),71)
IF P_P='Y'
   @0,0 SAY " "
   SET DEVICE TO SCREEN
ELSE
   DO H_SCR
ENDIF
RETURN
*!****************************************
*!    Procedure: NOWOLAND
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls
*!          : GET_REG (proc.in VFIPR.PRG)
*!          : H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!          : PDATA.DBF
*!          : PART101.DBF
*!          : PART106.DBF
*!
```

```
 *!   Indexes
 *!         : PDATA.NDX
 *!         : ID101.NDX
 *!*************************************
 PROC NOWOLAND
 S_S=0
 DO GET_REG
 IF S_S=0
    RETURN
 ENDIF
 SST=STR(S_S,1,0)
 SELE A
 USE PDATA INDEX PDATA
 SEEK SST
 POP1=S_P1
 POP2=S_P2
 POP3=S_P3
 POP4=S_P4
 USE PART101 INDEX ID101
 SELECT B
 USE PART106
 SET RELATION TO ID INTO A
 SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
 GO TOP
 STORE 0 TO T1,T2,T3,T4,N1,N2,N3,N4,
                              K1,K2,K3,K4,S1,S2,S3,S4
 DO WHILE .NOT. EOF()
      PEA=PEA1+PEA2+PEA3
      LEB=LEB1+LEB2+LEB3
      BAN=BAN1+BAN2+BAN3
      OTH=OTH1+OTH2+OTH3
      IF A->LAND='1'
           T1=T1+PEA
           N1=N1+LEB
           K1=K1+BAN
           S1=S1+OTH
      ENDIF
      IF A->LAND='2'
           T2=T2+PEA
           N2=N2+LEB
           K2=K2+BAN
           S2=S2+OTH
      ENDIF
      IF A->LAND='3'
           T3=T3+PEA
           N3=N3+LEB
           K3=K3+BAN
           S3=S3+OTH
      ENDIF
      IF A->LAND='4'
           T4=T4+PEA
           N4=N4+LEB
           K4=K4+BAN
           S4=S4+OTH
      ENDIF
      SKIP
 ENDDO
 T=T1+T2+T3+T4
 N=N1+N2+N3+N4
 K=K1+K2+K3+K4
 S=S1+S2+S3+S4
 PT1=T1/POP1
 PT2=T2/POP2
 PT3=T3/POP3
 PT4=T4/POP4
 PN1=N1/POP1
 PN2=N2/POP2
 PN3=N3/POP3
 PN4=N4/POP4
 PK1=K1/POP1
 PK2=K2/POP2
```

```
PK3=K3/POP3
PK4=K4/POP4
PS1=S1/POP1
PS2=S2/POP2
PS3=S3/POP3
PS4=S4/POP4
CLEAR
IF P_P='Y'
     SET DEVICE TO PRINT
ENDIF
@2,2 SAY 'Stock of trees of no wood value by
                       land ownership - Stratum '+SST
@3,2 SAY REPLICATE(CHR(196),63)
@4,2 SAY '                    Peara     Lebu
                                         Banana
@5,2 SAY '  Acres            no. pc    no. pc      Others'
                                          no. pc     no. pc'
@6,2 SAY ' -------          -------   -------
                                              ----     ---'
@7,2 SAY 'Less than .50'
@7,18 SAY T1 PICTURE '99999'
@7,24 SAY PT1 PICTURE '9.99'
@7,30 SAY N1 PICTURE '99999'
@7,36 SAY PN1 PICTURE '9.99'
@7,42 SAY K1 PICTURE '99999'
@7,48 SAY PK1 PICTURE '99.99'
@7,54 SAY S1 PICTURE '99999'
@7,60 SAY PS1 PICTURE '9.99'
@9,2 SAY '.50 to <2.50'
@9,18 SAY T2 PICTURE '99999'
@9,24 SAY PT2 PICTURE '9.99'
@9,30 SAY N2 PICTURE '99999'
@9,36 SAY PN2 PICTURE '9.99'
@9,42 SAY K2 PICTURE '99999'
@9,48 SAY PK2 PICTURE '99.99'
@9,54 SAY S2 PICTURE '99999'
@9,60 SAY PS2 PICTURE '9.99'
@11,2 SAY '2.50 to <7.50'
@11,18 SAY T3 PICTURE '99999'
@11,24 SAY PT3 PICTURE '9.99'
@11,30 SAY N3 PICTURE '99999'
@11,36 SAY PN3 PICTURE '9.99'
@11,42 SAY K3 PICTURE '99999'
@11,48 SAY PK3 PICTURE '99.99'
@11,54 SAY S3 PICTURE '99999'
@11,60 SAY PS3 PICTURE '9.99'
@13,2 SAY '7.50 & above'
@13,18 SAY T4 PICTURE '99999'
@13,24 SAY PT4 PICTURE '9.99'
@13,30 SAY N4 PICTURE '99999'
@13,36 SAY PN4 PICTURE '9.99'
@13,42 SAY K4 PICTURE '99999'
@13,48 SAY PK4 PICTURE '99.99'
@13,54 SAY S4 PICTURE '99999'
@13,60 SAY PS4 PICTURE '9.99'
@14,2 SAY REPLICATE('-',63)
@15,17 SAY T PICTURE '999999'
@15,29 SAY N PICTURE '999999'
@15,41 SAY K PICTURE '999999'
@15,53 SAY S PICTURE '999999'
@16,2 SAY REPLICATE(CHR(196),63)
@18,2 SAY 'no - number'
@19,2 SAY 'pc - per capita'
IF P_P='Y'
     @0,0 SAY " "
     SET DEVICE TO SCREEN
ELSE
     DO H_SCR
ENDIF
RETURN
*!*************************************
```

```
*!   Procedure: CANELAND
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls: H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!         : PDATA.DBF
*!         : PART101.DBF
*!   Indexes: PDATA.NDX
*!************************************
PROC CANELAND
SELE A
USE PDATA INDEX PDATA
SEEK 'T'
POP=S_PT
POP1=S_P1
POP2=S_P2
POP3=S_P3
POP4=S_P4
STORE 0 TO G1,G2,G3,G4,J1,J2,J3,
                                        J4,M1,M2,M3,M4
USE PART101
DO WHILE .NOT. EOF()
    IF LAND='1'
        G1=G1+CA_GOLLA
        J1=J1+CA_JALI
        M1=M1+CA_MURTA
    ENDIF
    IF LAND='2'
        G2=G2+CA_GOLLA
        J2=J2+CA_JALI
        M2=M2+CA_MURTA
    ENDIF
    IF LAND='3'
        G3=G3+CA_GOLLA
        J3=J3+CA_JALI
        M3=M3+CA_MURTA
    ENDIF
    IF LAND='4'
        G4=G4+CA_GOLLA
        J4=J4+CA_JALI
        M4=M4+CA_MURTA
    ENDIF
    SKIP
ENDDO
T1=G1+J1+M1
T2=G2+J2+M2
T3=G3+J3+M3
T4=G4+J4+M4
GT=G1+G2+G3+G4
JT=J1+J2+J3+J4
MT=M1+M2+M3+M4
CT=GT+JT+MT
PC1=T1/POP1
PC2=T2/POP2
PC3=T3/POP3
PC4=T4/POP4
GP=GT/POP
JP=JT/POP
MP=MT/POP
CP=CT/POP
CLEAR
IF P_P='Y'
    SET DEVICE TO PRINT
ENDIF
@2,5 SAY '          Canes: number of clums
                                        - All Strata'

@3,5 SAY REPLICATE(CHR(196),68)
@4,5 SAY 'Land in acres    Golla      Jali
                        Murta     Total     Per Capita'

@5,5 SAY '-------------    -----   ----   -----
                                       -----   ----------'

@6,5 SAY 'Less than .50'
@6,25 SAY G1 PICTURE '9999'
@6,35 SAY J1 PICTURE '9999'
@6,45 SAY M1 PICTURE '9999'
@6,55 SAY T1 PICTURE '9999'
@6,65 SAY PC1 PICTURE '.999'
@8,5 SAY '.50 to <2.50'
@8,25 SAY G2 PICTURE '9999'
@8,35 SAY J2 PICTURE '9999'
@8,45 SAY M2 PICTURE '9999'
@8,55 SAY T2 PICTURE '9999'
@8,65 SAY PC2 PICTURE '.999'
@10,5 SAY '2.50 to <7.5'
@10,25 SAY G3 PICTURE '9999'
@10,35 SAY J3 PICTURE '9999'
@10,45 SAY M3 PICTURE '9999'
@10,55 SAY T3 PICTURE '9999'
@10,65 SAY PC3 PICTURE '.999'
@12,5 SAY '7.50 & above'
@12,25 SAY G4 PICTURE '9999'
@12,35 SAY J4 PICTURE '9999'
@12,45 SAY M4 PICTURE '9999'
@12,55 SAY T4 PICTURE '9999'
@12,65 SAY PC4 PICTURE '.999'
@13,5 SAY REPLICATE('-',68)
@14,5 SAY 'Total'
@14,25 SAY GT PICTURE '9999'
@14,35 SAY JT PICTURE '9999'
@14,45 SAY MT PICTURE '9999'
@14,55 SAY CT PICTURE '9999'
@15,5 SAY REPLICATE('-',68)
@16,5 SAY 'Per Capita'
@16,25 SAY GP PICTURE '.999'
@16,35 SAY JP PICTURE '.999'
@16,45 SAY MP PICTURE '.999'
@16,55 SAY CP PICTURE '.999'
@17,5 SAY REPLICATE(CHR(196),68)
IF P_P='Y'
    @0,0 SAY " "
    SET DEVICE TO SCREEN
ELSE
    DO H_SCR
ENDIF
RETURN
*!************************************
*!   Procedure: FUELWHOS
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls: H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!         : PART101.DBF
*!         : PART201.DBF
*!   Indexes: ID101.NDX
*!************************************
PROC FUELWHOS
SELECT A
USE PART101 INDEX ID101
SELECT B
STORE 0 TO OW1,OW2,OW3,
                                        OW4,OW5,OW6
STORE 0 TO FM1,FM2,FM3,FM4,FM5,FM6
STORE 0 TO PS1,PS2,PS3,PS4,PS5,PS6
STORE 0 TO OT1,OT2,OT3,OT4,OT5,OT6
USE PART201
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
    IF SUBSTR(A->SAMPLE,1,1)='1'
        IF FU_B='1'
            OW1=OW1+1
        ENDIF
        IF FU_B='2'
            FM1=FM1+1
        ENDIF
        IF FU_B='3'
            PS1=PS1+1
        ENDIF
        IF FU_B='4'
            OT1=OT1+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='2'
        IF FU_B='1'
            OW2=OW2+1
        ENDIF
        IF FU_B='2'
            FM2=FM2+1
        ENDIF
        IF FU_B='3'
            PS2=PS2+1
        ENDIF
        IF FU_B='4'
            OT2=OT2+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='3'
        IF FU_B='1'
            OW3=OW3+1
        ENDIF
        IF FU_B='2'
            FM3=FM3+1
        ENDIF
        IF FU_B='3'
            PS3=PS3+1
        ENDIF
        IF FU_B='4'
            OT3=OT3+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='4'
        IF FU_B='1'
            OW4=OW4+1
        ENDIF
        IF FU_B='2'
            FM4=FM4+1
        ENDIF
        IF FU_B='3'
            PS4=PS4+1
        ENDIF
        IF FU_B='4'
            OT4=OT4+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='5'
        IF FU_B='1'
            OW5=OW5+1
        ENDIF
        IF FU_B='2'
            FM5=FM5+1
        ENDIF
        IF FU_B='3'
            PS5=PS5+1
        ENDIF
        IF FU_B='4'
            OT5=OT5+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='6'
        IF FU_B='1'
            OW6=OW6+1
        ENDIF
        IF FU_B='2'
            FM6=FM6+1
        ENDIF
        IF FU_B='3'
            PS6=PS6+1
        ENDIF
        IF FU_B='4'
            OT6=OT6+1
        ENDIF
    ENDIF
    SKIP
ENDDO
R1=OW1+FM1+PS1+OT1
R2=OW2+FM2+PS2+OT2
R3=OW3+FM3+PS3+OT3
R4=OW4+FM4+PS4+OT4
R5=OW5+FM5+PS5+OT5
R6=OW6+FM6+PS6+OT6
POW1=(OW1/R1)*100
PFM1=(FM1/R1)*100
PPS1=(PS1/R1)*100
POT1=(OT1/R1)*100
POW2=(OW2/R2)*100
PFM2=(FM2/R2)*100
PPS2=(PS2/R2)*100
POT2=(OT2/R2)*100
POW3=(OW3/R3)*100
PFM3=(FM3/R3)*100
PPS3=(PS3/R3)*100
POT3=(OT3/R3)*100
POW4=(OW4/R4)*100
PFM4=(FM4/R4)*100
PPS4=(PS4/R4)*100
POT4=(OT4/R4)*100
POW5=(OW5/R5)*100
PFM5=(FM5/R5)*100
PPS5=(PS5/R5)*100
POT5=(OT5/R5)*100
POW6=(OW6/R6)*100
PFM6=(FM6/R6)*100
PPS6=(PS6/R6)*100
POT6=(OT6/R6)*100
RT=R1+R2+R3+R4+R5+R6
OWT=OW1+OW2+OW3+OW4+OW5+OW6
FMT=FM1+FM2+FM3+FM4+FM5+FM6
PST=PS1+PS2+PS3+PS4+PS5+PS6
OTT=OT1+OT2+OT3+OT4+OT5+OT6
POWT=(OWT/RT)*100
PFMT=(FMT/RT)*100
PPST=(PST/RT)*100
POTT=(OTT/RT)*100
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '           Who collects fuel materials
                                 in rural Bangladesh'
@3,2 SAY REPLICATE(CHR(196),69)
@4,2 SAY 'Strata          1      2      3      4
                                   5      6      Total'
@5,2 SAY REPLICATE('-',69)
@6,2 SAY 'Response no.'
@6,17 SAY R1 PICTURE '9999'
@6,25 SAY R2 PICTURE '9999'
@6,33 SAY R3 PICTURE '9999'
@6,41 SAY R4 PICTURE '9999'
@6,49 SAY R5 PICTURE '9999'
@6,57 SAY R6 PICTURE '9999'
@6,66 SAY RT PICTURE '9999'
@7,2 SAY REPLICATE('-',69)
@8,2 SAY '                %      %      %             %'
                             %      %      %
@10,2 SAY 'Owner'
@10,18 SAY POW1 PICTURE '99.9'
@10,26 SAY POW2 PICTURE '99.9'
```

```
@10,34 SAY POW3 PICTURE '99.9'
@10,42 SAY POW4 PICTURE '99.9'
@10,50 SAY POW5 PICTURE '99.9'
@10,58 SAY POW6 PICTURE '99.9'
@10,66 SAY POWT PICTURE '99.9'
@12,2 SAY 'Family member'
@12,18 SAY PFM1 PICTURE '99.9'
@12,26 SAY PFM2 PICTURE '99.9'
@12,34 SAY PFM3 PICTURE '99.9'
@12,42 SAY PFM4 PICTURE '99.9'
@12,50 SAY PFM5 PICTURE '99.9'
@12,58 SAY PFM6 PICTURE '99.9'
@12,66 SAY PFMT PICTURE '99.9'
@14,2 SAY 'Paid servant'
@14,18 SAY PPS1 PICTURE '99.9'
@14,26 SAY PPS2 PICTURE '99.9'
@14,34 SAY PPS3 PICTURE '99.9'
@14,42 SAY PPS4 PICTURE '99.9'
@14,50 SAY PPS5 PICTURE '99.9'
@14,58 SAY PPS6 PICTURE '99.9'
@14,66 SAY PPST PICTURE '99.9'
@16,2 SAY 'Others'
@16,18 SAY POT1 PICTURE '99.9'
@16,26 SAY POT2 PICTURE '99.9'
@16,34 SAY POT3 PICTURE '99.9'
@16,42 SAY POT4 PICTURE '99.9'
@16,50 SAY POT5 PICTURE '99.9'
@16,58 SAY POT6 PICTURE '99.9'
@16,66 SAY POTT PICTURE '99.9'
@17,2 SAY REPLICATE(CHR(196),69)
IF P_P='Y'
    @0,0 SAY " "
    SET DEVICE TO SCREEN
ELSE
    DO H_SCR
ENDIF
RETURN
*!****************************************
*!   Procedure: STOCSMAL
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls
*!         : GET_REG (proc.in VFIPR.PRG)
*!         : H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!         : PDATA.DBF
*!         : PART101.DBF
*!         : PART103.DBF
*!   Indexes
*!         : PDATA.NDX
*!         : ID101.NDX
*!         : CODE103.NDX
*!****************************************
PROC STOCSMAL
SELECT A
S_S=0
DO GET_REG
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
SP=S_PT
TP=T_POP
USE PART101 INDEX ID101
SELECT B
USE PART103 INDEX CODE103
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP
```

```
TOT=0
TPC=0
LN=5
CLEAR
IF P_P='Y'
    SET DEVICE TO PRINT
ENDIF
@1,2 SAY 'Stock of trees upto 8" dia(BH)
                          & greater than 5ft height'
@2,2 SAY '
                     . ( Stratum '+
                                           SST+' )'
@3,2 SAY REPLICATE(CHR(196),56)
@4,7 SAY 'ID      Species      Stems(000)
                                         Per Capita'
@5,7 SAY '--      -------      ----------
                                         ----------'
DO WHILE .NOT. EOF()
    CD=CODE
    NO=0
    LN=LN+1
    DO WHILE CODE=CD
        NO=NO+NUMBER
        SKIP
    ENDDO
    PC=NO/SP
    NO=TP*PC/1000
    TOT=TOT+NO
    TPC=TPC+PC
    IF CD='01'
        X='Mango'
    ENDIF
    IF CD='02'
        X='Jack'
    ENDIF
    IF CD='03'
        X='Rain'
    ENDIF
    IF CD='04'
        X='Simul'
    ENDIF
    IF CD='05'
        X='Bat'
    ENDIF
    IF CD='06'
        X='Madar'
    ENDIF
    IF CD='07'
        X='Koroi'
    ENDIF
    IF CD='08'
        X='Chakua'
    ENDIF
    IF CD='09'
        X='Jam'
    ENDIF
    IF CD='10'
        X='Jiul'
    ENDIF
    IF CD='11'
        X='Gab'
    ENDIF
    IF CD='12'
        X='Tetul'
    ENDIF
    IF CD='13'
        X='Bel'
    ENDIF
    IF CD='14'
        X='Pitali'
    ENDIF
    IF CD='15'
```

```
        X='Chhatim'
    ENDIF
    IF CD='16'
        X='Kadam'
    ENDIF
    IF CD='17'
        X='Debdaru'
    ENDIF
    IF CD='18'
        X='Jarul'
    ENDIF
    IF CD='19'
        X='Sal'
    ENDIF
    IF CD='20'
        X='Segun'
    ENDIF
    IF CD='21'
        X='Garjan'
    ENDIF
    IF CD='22'
        X='Palash'
    ENDIF
    IF CD='23'
        X='Lichu'
    ENDIF
    IF CD='24'
        X='Others'
    ENDIF
    @LN,7 SAY CD
    @LN,17 SAY X
    @LN,33 SAY NO PICTURE '999999'
    @LN,47 SAY PC PICTURE '99.99'
    IF P_P='N'.AND.LN=20.AND.
                                        .NOT. EOF()
        DO H_SCR
        @6,0 CLEAR TO 24,79
        LN=5
    ENDIF
ENDDO
LN=LN+1
@LN,7 SAY REPLICATE('-',45)
@LN+1,17 SAY 'All species:'
@LN+1,33 SAY TOT PICTURE '999999'
@LN+1,47 SAY TPC PICTURE '99.99'
@LN+2,2 SAY REPLICATE(CHR(196),56)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!*************************************
*!   Procedure: STOCBAM
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls
*!       : GET_REG (proc.in VFIPR.PRG)
*!       : H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!       : PDATA.DBF
*!       : PART101.DBF
*!       : PART102.DBF
*!   Indexes
*!       : PDATA.NDX
*!       : ID101.NDX
*!*************************************

PROC STOCBAM
S_S=0
DO GET_REG
```

```
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
POP=S_PT
USE PART101 INDEX ID101
SELECT B
STORE 0 TO L11,L12,L13,L21,L22,L23,
                         L31,L32,L33,L41,L42,L43

STORE 0 TO L51,L52,L53,L61,L62,L63,L71,
                            L72,L73,L81,L82,L83

STORE 0 TO L91,L92,L93,L101,L102,L103,
                               L111,L112,L113

STORE 0 TO L121,L122,L123
USE PART102
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP
DO WHILE .NOT. EOF()
    L11=L11+B1_CL
    L12=L12+B1_MS
    L13=L13+B1_IS
    L21=L21+B2_CL
    L22=L22+B2_MS
    L23=L23+B2_IS
    L31=L31+B3_CL
    L32=L32+B3_MS
    L33=L33+B3_IS
    L41=L41+B4_CL
    L42=L42+B4_MS
    L43=L43+B4_IS
    L51=L51+B5_CL
    L52=L52+B5_MS
    L53=L53+B5_IS
    L61=L61+B6_CL
    L62=L62+B6_MS
    L63=L63+B6_IS
    L71=L71+B7_CL
    L72=L72+B7_MS
    L73=L73+B7_IS
    L81=L81+B8_CL
    L82=L82+B8_MS
    L83=L83+B8_IS
    L91=L91+B9_CL
    L92=L92+B9_MS
    L93=L93+B9_IS
    L101=L101+B10_CL
    L102=L102+B10_MS
    L103=L103+B10_IS
    L111=L111+B11_CL
    L112=L112+B11_MS
    L113=L113+B11_IS
    L121=L121+B12_CL
    L122=L122+B12_MS
    L123=L123+B12_IS
    SKIP
ENDDO
IT=1
STORE 0 TO TTON,TMC,TIC
LN=6
CLEAR
IF P_P='Y'
    SET DEVICE TO PRINT
ENDIF
@1,2 SAY '                    BAMBOO
                    RESOURCES - Stratum '+SST
@2,2 SAY REPLICATE(CHR(196),65)
@3,2 SAY '                      Mature Culms
```

```
                                           Immature Culms'
@4,2 SAY 'Species             Tons     No.
                         No/ Capita        No.   No/ Capita'
@5,2 SAY '-------           -----  -----   -----
                                                 -----    -----'
DO WHILE IT<=12
   IF IT=1
      NAM='Katabash'
      TON=L12/ 63
      MC=L12
      PMC=MC/ POP
      IC=L13
      PIC=IC/ POP
   ENDIF
   IF IT=2
      NAM='Bariala'
      TON=L22/ 77
      MC=L22
      PMC=MC/ POP
      IC=L23
      PIC=IC/ POP
   ENDIF
   IF IT=3
      NAM='Barua'
      TON=L32/ 115
      MC=L32
      PMC=MC/ POP
      IC=L33
      PIC=IC/ POP
   ENDIF
   IF IT=4
      NAM='Jai'
      TON=L42/ 126
      MC=L42
      PMC=MC/ POP
      IC=L43
      PIC=IC/ POP
   ENDIF
   IF IT=5
      NAM='Makhal'
      TON=L52/ 170
      MC=L52
      PMC=MC/ POP
      IC=L53
      PIC=IC/ POP
   ENDIF
   IF IT=6
      NAM='Orah'
      TON=L62/ 202
      MC=L62
      PMC=MC/ POP
      IC=L63
      PIC=IC/ POP
   ENDIF
   IF IT=7
      NAM='Kaliseri'
      TON=L72/ 213
      MC=L72
      PMC=MC/ POP
      IC=L73
      PIC=IC/ POP
   ENDIF
   IF IT=8
      NAM='Tarala'
      TON=L82/ 233
      MC=L82
      PMC=MC/ POP
      IC=L83
      PIC=IC/ POP
   ENDIF
   IF IT=9
      NAM='Barak'
      TON=L92/ 77
      MC=L92
      PMC=MC/ POP
      IC=L93
      PIC=IC/ POP
   ENDIF
   IF IT=10
      NAM='Mitinga'
      TON=L102/ 450
      MC=L102
      PMC=MC/ POP
      IC=L103
      PIC=IC/ POP
   ENDIF
   IF IT=11
      NAM='Muli'
      TON=L112/ 450
      MC=L112
      PMC=MC/ POP
      IC=L113
      PIC=IC/ POP
   ENDIF
   IF IT=12
      NAM='Others'
      TON=L122/ 488
      MC=L122
      PMC=MC/ POP
      IC=L123
      PIC=IC/ POP
   ENDIF
   TTON=TTON+TON
   TMC=TMC+MC
   TPMC=TMC/ POP
   TIC=TIC+IC
   TPIC=TIC/ POP
   @LN,2 SAY NAM
   @LN,20 SAY TON PICTURE '99999'
   @LN,27 SAY MC PICTURE '999999'
   @LN,38 SAY PMC PICTURE '99.99'
   @LN,49 SAY IC PICTURE '999999'
   @LN,60 SAY PIC PICTURE '99.99'
   LN=LN+1
   IT=IT+1
ENDDO
@LN,2 SAY REPLICATE('-',64)
@LN+1,9 SAY 'Total:'
@LN+1,20 SAY TTON PICTURE '99999'
@LN+1,27 SAY TMC PICTURE '999999'
@LN+1,38 SAY TPMC PICTURE '99.99'
@LN+1,49 SAY TIC PICTURE '999999'
@LN+1,60 SAY TPIC PICTURE '99.99'
@LN+2,2 SAY REPLICATE(CHR(196),65)
IF P_P='Y'
   SET DEVICE TO SCREEN
   @0,0 SAY " "
ELSE
   DO H_SCR
ENDIF
RETURN

*|****************************************
*|     Procedure: FUELFAR
*|     Called by: PRNT (proc.in VFIPR.PRG)
*|     Calls: H_SCR (proc.in VFIPR.PRG)
*|     Uses
*|          : PART101.DBF
*|          : PART201.DBF
*|     Indexes: ID101.NDX
*|****************************************
PROC FUELFAR
```

```
SELECT A
USE PART101 INDEX ID101
SELECT B
STORE 0 TO OW1,OW2,OW3,OW4,OW5,OW6
STORE 0 TO FM1,FM2,FM3,FM4,FM5,FM6
STORE 0 TO PS1,PS2,PS3,PS4,PS5,PS6
USE PART201
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
    IF SUBSTR(A->SAMPLE,1,1)='1'
        IF FU_D='1'
            OW1=OW1+1
        ENDIF
        IF FU_D='2'
            FM1=FM1+1
        ENDIF
        IF FU_D='3'
            PS1=PS1+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='2'
        IF FU_D='1'
            OW2=OW2+1
        ENDIF
        IF FU_D='2'
            FM2=FM2+1
        ENDIF
        IF FU_D='3'
            PS2=PS2+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='3'
        IF FU_D='1'
            OW3=OW3+1
        ENDIF
        IF FU_D='2'
            FM3=FM3+1
        ENDIF
        IF FU_D='3'
            PS3=PS3+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='4'
        IF FU_D='1'
            OW4=OW4+1
        ENDIF
        IF FU_D='2'
            FM4=FM4+1
        ENDIF
        IF FU_D='3'
            PS4=PS4+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='5'
        IF FU_D='1'
            OW5=OW5+1
        ENDIF
        IF FU_D='2'
            FM5=FM5+1
        ENDIF
        IF FU_D='3'
            PS5=PS5+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='6'
        IF FU_D='1'
            OW6=OW6+1
        ENDIF
        IF FU_D='2'
            FM6=FM6+1
        ENDIF
        IF FU_D='3'
            PS6=PS6+1
        ENDIF
    ENDIF
    SKIP
ENDDO
R1=OW1+FM1+PS1
R2=OW2+FM2+PS2
R3=OW3+FM3+PS3
R4=OW4+FM4+PS4
R5=OW5+FM5+PS5
R6=OW6+FM6+PS6
POW1=(OW1/R1)*100
PFM1=(FM1/R1)*100
PPS1=(PS1/R1)*100
POW2=(OW2/R2)*100
PFM2=(FM2/R2)*100
PPS2=(PS2/R2)*100
POW3=(OW3/R3)*100
PFM3=(FM3/R3)*100
PPS3=(PS3/R3)*100
POW4=(OW4/R4)*100
PFM4=(FM4/R4)*100
PPS4=(PS4/R4)*100
POW5=(OW5/R5)*100
PFM5=(FM5/R5)*100
PPS5=(PS5/R5)*100
POW6=(OW6/R6)*100
PFM6=(FM6/R6)*100
PPS6=(PS6/R6)*100
RT=R1+R2+R3+R4+R5+R6
OWT=OW1+OW2+OW3+OW4+OW5+OW6
FMT=FM1+FM2+FM3+FM4+FM5+FM6
PST=PS1+PS2+PS3+PS4+PS5+PS6
POWT=(OWT/RT)*100
PFMT=(FMT/RT)*100
PPST=(PST/RT)*100
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '    How far one has to go to collect
                              fuel in rural Bangladesh'
@3,2 SAY REPLICATE(CHR(196),69)
@4,2 SAY 'Strata          1     2     3     4
                                  5     6     Total'
@5,2 SAY REPLICATE('-',69)
@6,2 SAY 'Response no.'
@6,17 SAY R1 PICTURE '9999'
@6,25 SAY R2 PICTURE '9999'
@6,33 SAY R3 PICTURE '9999'
@6,41 SAY R4 PICTURE '9999'
@6,49 SAY R5 PICTURE '9999'
@6,57 SAY R6 PICTURE '9999'
@6,66 SAY RT PICTURE '9999'
@7,2 SAY REPLICATE('-',69)
@8,2 SAY '                %     %     %
                            %     %     %     %'
@10,2 SAY '< 1 mile'
@10,18 SAY POW1 PICTURE '99.9'
@10,26 SAY POW2 PICTURE '99.9'
@10,34 SAY POW3 PICTURE '99.9'
@10,42 SAY POW4 PICTURE '99.9'
@10,50 SAY POW5 PICTURE '99.9'
@10,58 SAY POW6 PICTURE '99.9'
@10,66 SAY POWT PICTURE '99.9'
@12,2 SAY '1 - 3 miles'
@12,18 SAY PFM1 PICTURE '99.9'
@12,26 SAY PFM2 PICTURE '99.9'
@12,34 SAY PFM3 PICTURE '99.9'
@12,42 SAY PFM4 PICTURE '99.9'
```

```
@12,50 SAY PFM5 PICTURE '99.9'
@12,58 SAY PFM6 PICTURE '99.9'
@12,66 SAY PFMT PICTURE '99.9'
@14,2 SAY '> 3 miles'
@14,18 SAY PPS1 PICTURE '99.9'
@14,26 SAY PPS2 PICTURE '99.9'
@14,34 SAY PPS3 PICTURE '99.9'
@14,42 SAY PPS4 PICTURE '99.9'
@14,50 SAY PPS5 PICTURE '99.9'
@14,58 SAY PPS6 PICTURE '99.9'
@14,66 SAY PPST PICTURE '99.9'
@15,2 SAY REPLICATE(CHR(196),69)
IF P_P='Y'
     SET DEVICE TO SCREEN
     @0,0 SAY " "
ELSE
     DO H_SCR
ENDIF
RETURN
*!**************************************
*!   Procedure: FUELAVAI
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls: H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!         : PART101.DBF
*!         : PART201.DBF
*!   Indexes: ID101.NDX
*!**************************************
PROC FUELAVAI
SELECT A
USE PART101 INDEX ID101
SELECT B
STORE 0 TO OW1,OW2,OW3,OW4,OW5,OW6
STORE 0 TO FM1,FM2,FM3,FM4,FM5,FM6
STORE 0 TO PS1,PS2,PS3,PS4,PS5,PS6
STORE 0 TO OT1,OT2,OT3,OT4,OT5,OT6
USE PART201
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
     IF SUBSTR(A->SAMPLE,1,1)='1'
          IF FU_F='1'
               OW1=OW1+1
          ENDIF
          IF FU_F='2'
               FM1=FM1+1
          ENDIF
          IF FU_F='3'
               PS1=PS1+1
          ENDIF
          IF FU_F='4'
               OT1=OT1+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='2'
          IF FU_F='1'
               OW2=OW2+1
          ENDIF
          IF FU_F='2'
               FM2=FM2+1
          ENDIF
          IF FU_F='3'
               PS2=PS2+1
          ENDIF
          IF FU_F='4'
               OT2=OT2+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='3'
          IF FU_F='1'
               OW3=OW3+1
          ENDIF
          IF FU_F='2'
               FM3=FM3+1
          ENDIF
          IF FU_F='3'
               PS3=PS3+1
          ENDIF
          IF FU_F='4'
               OT3=OT3+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='4'
          IF FU_F='1'
               OW4=OW4+1
          ENDIF
          IF FU_F='2'
               FM4=FM4+1
          ENDIF
          IF FU_F='3'
               PS4=PS4+1
          ENDIF
          IF FU_F='4'
               OT4=OT4+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='5'
          IF FU_F='1'
               OW5=OW5+1
          ENDIF
          IF FU_F='2'
               FM5=FM5+1
          ENDIF
          IF FU_F='3'
               PS5=PS5+1
          ENDIF
          IF FU_F='4'
               OT5=OT5+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='6'
          IF FU_F='1'
               OW6=OW6+1
          ENDIF
          IF FU_F='2'
               FM6=FM6+1
          ENDIF
          IF FU_F='3'
               PS6=PS6+1
          ENDIF
          IF FU_F='4'
               OT6=OT6+1
          ENDIF
     ENDIF
     SKIP
ENDDO
R1=OW1+FM1+PS1+OT1
R2=OW2+FM2+PS2+OT2
R3=OW3+FM3+PS3+OT3
R4=OW4+FM4+PS4+OT4
R5=OW5+FM5+PS5+OT5
R6=OW6+FM6+PS6+OT6
POW1=(OW1/ R1)*100
PFM1=(FM1/ R1)*100
PPS1=(PS1/ R1)*100
POT1=(OT1/ R1)*100
POW2=(OW2/ R2)*100
PFM2=(FM2/ R2)*100
PPS2=(PS2/ R2)*100
POT2=(OT2/ R2)*100
POW3=(OW3/ R3)*100
PFM3=(FM3/ R3)*100
PPS3=(PS3/ R3)*100
```

```
POT3=(OT3/R3)*100
POW4=(OW4/R4)*100
PFM4=(FM4/R4)*100
PPS4=(PS4/R4)*100
POT4=(OT4/R4)*100
POW5=(OW5/R5)*100
PFM5=(FM5/R5)*100
PPS5=(PS5/R5)*100
POT5=(OT5/R5)*100
POW6=(OW6/R6)*100
PFM6=(FM6/R6)*100
PPS6=(PS6/R6)*100
POT6=(OT6/R6)*100
RT=R1+R2+R3+R4+R5+R6
OWT=OW1+OW2+OW3+OW4+OW5+OW6
FMT=FM1+FM2+FM3+FM4+FM5+FM6
PST=PS1+PS2+PS3+PS4+PS5+PS6
OTT=OT1+OT2+OT3+OT4+OT5+OT6
POWT=(OWT/RT)*100
PFMT=(FMT/RT)*100
PPST=(PST/RT)*100
POTT=(OTT/RT)*100
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '  How the repondants ranked
                    availability of fuel in their area'
@3,2 SAY REPLICATE(CHR(196),69)
@4,2 SAY 'Strata          1       2       3       4
                                   5       6       Total'
@5,2 SAY REPLICATE('-',69)
@6,2 SAY 'Response no.'
@6,17 SAY R1 PICTURE '9999'
@6,25 SAY R2 PICTURE '9999'
@6,33 SAY R3 PICTURE '9999'
@6,41 SAY R4 PICTURE '9999'
@6,49 SAY R5 PICTURE '9999'
@6,57 SAY R6 PICTURE '9999'
@6,66 SAY RT PICTURE '9999'
@7,2 SAY REPLICATE('-',69)
@8,2 SAY '               %       %       %
                                   %       %       %       %'
@10,2 SAY 'Scanty'
@10,18 SAY POW1 PICTURE '99.9'
@10,26 SAY POW2 PICTURE '99.9'
@10,34 SAY POW3 PICTURE '99.9'
@10,42 SAY POW4 PICTURE '99.9'
@10,50 SAY POW5 PICTURE '99.9'
@10,58 SAY POW6 PICTURE '99.9'
@10,66 SAY POWT PICTURE '99.9'
@12,2 SAY 'Ample'
@12,18 SAY PFM1 PICTURE '99.9'
@12,26 SAY PFM2 PICTURE '99.9'
@12,34 SAY PFM3 PICTURE '99.9'
@12,42 SAY PFM4 PICTURE '99.9'
@12,50 SAY PFM5 PICTURE '99.9'
@12,58 SAY PFM6 PICTURE '99.9'
@12,66 SAY PFMT PICTURE '99.9'
@14,2 SAY 'Abundant'
@14,18 SAY PPS1 PICTURE '99.9'
@14,26 SAY PPS2 PICTURE '99.9'
@14,34 SAY PPS3 PICTURE '99.9'
@14,42 SAY PPS4 PICTURE '99.9'
@14,50 SAY PPS5 PICTURE '99.9'
@14,58 SAY PPS6 PICTURE '99.9'
@14,66 SAY PPST PICTURE '99.9'
@16,2 SAY 'Not known'
@16,18 SAY POT1 PICTURE '99.9'
@16,26 SAY POT2 PICTURE '99.9'
@16,34 SAY POT3 PICTURE '99.9'
@16,42 SAY POT4 PICTURE '99.9'
@16,50 SAY POT5 PICTURE '99.9'
@16,58 SAY POT6 PICTURE '99.9'
@16,66 SAY POTT PICTURE '99.9'
@17,2 SAY REPLICATE(CHR(196),69)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!*************************************
*!      Procedure: STOCNOWO
*!      Called by: PRNT (proc.in VFIPR.PRG)
*!      Calls: H_SCR (proc.in VFIPR.PRG)
*!      Uses
*!            : PDATA.DBF
*!            : PART101.DBF
*!            : PART106.DBF
*!      Indexes
*!            : PDATA.NDX
*!            : ID101.NDX
*!*************************************
PROC STOCNOWO
SELECT A
USE PDATA INDEX PDATA
SEEK 'T'
POP=S_PT
USE PART101 INDEX ID101
SELECT B
STORE 0 TO T1,T2,T3,N1,N2,N3,K1,K2,
                        K3,S1,S2,S3,TOT,PTOT
USE PART106
SET RELATION TO ID INTO A
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@1,2 SAY '          Stock of tree of no wood
                              value - All Strata'
@2,2 SAY REPLICATE(CHR(196),63)
@3,2 SAY '        Number according to
                                   height in ft.'
@4,2 SAY 'Local    -------------------------
                        ----                Per'
@5,2 SAY 'Name     Below 5      5 - 10
                   Above 10    Total     Capita'
@6,2 SAY '----    -------    ------   --------
                                  -----    -----'
DO WHILE .NOT. EOF()
    T1=T1+PEA1
    T2=T2+PEA2
    T3=T3+PEA3
    N1=N1+LEB1
    N2=N2+LEB2
    N3=N3+LEB3
    K1=K1+BAN1
    K2=K2+BAN2
    K3=K3+BAN3
    S1=S1+OTH1
    S2=S2+OTH2
    S3=S3+OTH3
    SKIP
ENDDO
T=T1+T2+T3
N=N1+N2+N3
K=K1+K2+K3
S=S1+S2+S3
TOT=T+N+K+S
PT=T/POP
```

```
PN=N/POP
PK=K/POP
PS=S/POP
PTOT=TOT/POP
@7,2 SAY 'Peara'
@7,11 SAY T1 PICTURE '999999'
@7,23 SAY T2 PICTURE '999999'
@7,35 SAY T3 PICTURE '999999'
@7,47 SAY T PICTURE '999999'
@7,59 SAY PT PICTURE '99.99'
@9,2 SAY 'Lebu'
@9,11 SAY N1 PICTURE '999999'
@9,23 SAY N2 PICTURE '999999'
@9,35 SAY N3 PICTURE '999999'
@9,47 SAY N PICTURE '999999'
@9,59 SAY PN PICTURE '99.99'
@11,2 SAY 'Banana'
@11,11 SAY K1 PICTURE '999999'
@11,23 SAY K2 PICTURE '999999'
@11,35 SAY K3 PICTURE '999999'
@11,47 SAY K PICTURE '999999'
@11,59 SAY PK PICTURE '99.99'
@13,2 SAY 'Others'
@13,11 SAY S1 PICTURE '999999'
@13,23 SAY S2 PICTURE '999999'
@13,35 SAY S3 PICTURE '999999'
@13,47 SAY S PICTURE '999999'
@13,59 SAY PS PICTURE '99.99'
@14,2 SAY REPLICATE('-',63)
@15,2 SAY 'Total for all types:'
@15,47 SAY TOT PICTURE '999999'
@15,59 SAY PTOT PICTURE '99.99'
@16,2 SAY REPLICATE(CHR(196),63)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!****************************************
*!   Procedure: BAMLAND
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls
*!          : GET_REG (proc.in VFIPR.PRG)
*!          : H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!          : PDATA.DBF
*!          : PART101.DBF
*!          : PART102.DBF
*!   Indexes
*!          : PDATA.NDX
*!          : ID101.NDX
*!****************************************
PROC BAMLAND
S_S=0
DO GET_REG
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
POP1=S_P1
POP2=S_P2
POP3=S_P3
POP4=S_P4
USE PART101 INDEX ID101
SELECT B
USE PART102
```

```
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP
STORE 0 TO L12A,L13A,L22A,L23A,
               L32A,L33A,L42A,L43A,L52A,L53A,
               L62A,L63A,L72A,L73A,L82A,L83A,
                 L92A,L93A,L102A,L103A,L112A,
                            L113A,L122A,L123A
STORE 0 TO L12B,L13B,L22B,L23B,L32B,
           L33B,L42B,L43B,L52B,L53B,L62B,L63B,
          L72B,L73B,L82B,L83B,L92B,L93B,L102B,
                L103B,L112B,L113B,L122B,L123B
STORE 0 TO L12C,L13C,L22C,L23C,L32C,
           L33C,L42C,L43C,L52C,L53C,L62C,L63C,
          L72C,L73C,L82C,L83C,L92C,L93C,L102C,
                L103C,L112C,L113C,L122C,L123C
STORE 0 TO L12D,L13D,L22D,L23D,L32D,
          L33D,L42D,L43D,L52D,L53D,L62D,L63D,
               L72D,L73D,L82D,L83D,L92D,L93D,
          L102D,L103D,L112D,L113D,L122D,L123D
DO WHILE .NOT. EOF()
     IF A->LAND='1'
          L12A=L12A+B1_MS
          L13A=L13A+B1_IS
          L22A=L22A+B2_MS
          L23A=L23A+B2_IS
          L32A=L32A+B3_MS
          L33A=L33A+B3_IS
          L42A=L42A+B4_MS
          L43A=L43A+B4_IS
          L52A=L52A+B5_MS
          L53A=L53A+B5_IS
          L62A=L62A+B6_MS
          L63A=L63A+B6_IS
          L72A=L72A+B7_MS
          L73A=L73A+B7_IS
          L82A=L82A+B8_MS
          L83A=L83A+B8_IS
          L92A=L92A+B9_MS
          L93A=L93A+B9_IS
          L102A=L102A+B10_MS
          L103A=L103A+B10_IS
          L112A=L112A+B11_MS
          L113A=L113A+B11_IS
          L122A=L122A+B12_MS
          L123A=L123A+B12_IS
     ENDIF
     IF A->LAND='2'
          L12B=L12B+B1_MS
          L13B=L13B+B1_IS
          L22B=L22B+B2_MS
          L23B=L23B+B2_IS
          L32B=L32B+B3_MS
          L33B=L33B+B3_IS
          L42B=L42B+B4_MS
          L43B=L43B+B4_IS
          L52B=L52B+B5_MS
          L53B=L53B+B5_IS
          L62B=L62B+B6_MS
          L63B=L63B+B6_IS
          L72B=L72B+B7_MS
          L73B=L73B+B7_IS
          L82B=L82B+B8_MS
          L83B=L83B+B8_IS
          L92B=L92B+B9_MS
          L93B=L93B+B9_IS
          L102B=L102B+B10_MS
          L103B=L103B+B10_IS
          L112B=L112B+B11_MS
          L113B=L113B+B11_IS
          L122B=L122B+B12_MS
```

```
        L123B=L123B+B12_IS
    ENDIF
    IF A->LAND='3'
        L12C=L12C+B1_MS
        L13C=L13C+B1_IS
        L22C=L22C+B2_MS
        L23C=L23C+B2_IS
        L32C=L32C+B3_MS
        L33C=L33C+B3_IS
        L42C=L42C+B4_MS
        L43C=L43C+B4_IS
        L52C=L52C+B5_MS
        L53C=L53C+B5_IS
        L62C=L62C+B6_MS
        L63C=L63C+B6_IS
        L72C=L72C+B7_MS
        L73C=L73C+B7_IS
        L82C=L82C+B8_MS
        L83C=L83C+B8_IS
        L92C=L92C+B9_MS
        L93C=L93C+B9_IS
        L102C=L102C+B10_MS
        L103C=L103C+B10_IS
        L112C=L112C+B11_MS
        L113C=L113C+B11_IS
        L122C=L122C+B12_MS
        L123C=L123C+B12_IS
    ENDIF
    IF A->LAND='4'
        L12D=L12D+B1_MS
        L13D=L13D+B1_IS
        L22D=L22D+B2_MS
        L23D=L23D+B2_IS
        L32D=L32D+B3_MS
        L33D=L33D+B3_IS
        L42D=L42D+B4_MS
        L43D=L43D+B4_IS
        L52D=L52D+B5_MS
        L53D=L53D+B5_IS
        L62D=L62D+B6_MS
        L63D=L63D+B6_IS
        L72D=L72D+B7_MS
        L73D=L73D+B7_IS
        L82D=L82D+B8_MS
        L83D=L83D+B8_IS
        L92D=L92D+B9_MS
        L93D=L93D+B9_IS
        L102D=L102D+B10_MS
        L103D=L103D+B10_IS
        L112D=L112D+B11_MS
        L113D=L113D+B11_IS
        L122D=L122D+B12_MS
        L123D=L123D+B12_IS
    ENDIF
    SKIP
ENDDO
A1=L12A+L22A+L32A+L42A+L52A+
                          L62A+L72A+L82A+L92A+
                               L102A+L112A+L122A
A2=L13A+L23A+L33A+L43A+L53A+
                          L63A+L73A+L83A+L93A+
                               L103A+L113A+L123A
B1=L12B+L22B+L32B+L42B+L52B+
                          L62B+L72B+L82B+L92B+
                               L102B+L112B+L122B
B2=L13B+L23B+L33B+L43B+L53B+
                          L63B+L73B+L83B+L93B+
                               L103B+L113B+L123B
C1=L12C+L22C+L32C+L42C+L52C+
                          L62C+L72C+L82C+L92C+
                               L102C+L112C+L122C
```

```
C2=L13C+L23C+L33C+L43C+L53C+
                          L63C+L73C+L83C+L93C+
                               L103C+L113C+L123C


D1=L12D+L22D+L32D+L42D+L52D+
                          L62D+L72D+L82D+L92D+
                               L102D+L112D+L122D
D2=L13D+L23D+L33D+L43D+L53D+
                          L63D+L73D+L83D+L93D+
                               L103D+L113D+L123D

PC1=A1/POP1
PC11=A2/POP1
PC2=B1/POP2
PC22=B2/POP2
PC3=C1/POP3
PC33=C2/POP3
PC4=D1/POP4
PC44=D2/POP4
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@0,2 SAY '     Bamboo resources by land
                 ownership in acres - Stratum '+SST
@1,2 SAY REPLICATE(CHR(196),70)
@2,2 SAY '             less than .5   .5 to <2.5
                              2.5 to <7.5   7.5 & above'
@3,2 SAY '             -----------  -----------
                                 -----------  ----------'
@4,2 SAY 'Species      mc     ic     mc     ic
                               mc     ic     mc     ic'
@5,2 SAY REPLICATE('-',70)
@6,2 SAY 'Katabash'
@6,14 SAY L12A PICTURE '99999'
@6,21 SAY L13A PICTURE '99999'
@6,29 SAY L12B PICTURE '99999'
@6,36 SAY L13B PICTURE '99999'
@6,44 SAY L12C PICTURE '99999'
@6,51 SAY L13C PICTURE '99999'
@6,59 SAY L12D PICTURE '99999'
@6,66 SAY L13D PICTURE '99999'
@7,2 SAY 'Bariala'
@7,14 SAY L22A PICTURE '99999'
@7,21 SAY L23A PICTURE '99999'
@7,29 SAY L22B PICTURE '99999'
@7,36 SAY L23B PICTURE '99999'
@7,44 SAY L22C PICTURE '99999'
@7,51 SAY L23C PICTURE '99999'
@7,59 SAY L22D PICTURE '99999'
@7,66 SAY L23D PICTURE '99999'
@8,2 SAY 'Barua'
@8,14 SAY L32A PICTURE '99999'
@8,21 SAY L33A PICTURE '99999'
@8,29 SAY L32B PICTURE '99999'
@8,36 SAY L33B PICTURE '99999'
@8,44 SAY L32C PICTURE '99999'
@8,51 SAY L33C PICTURE '99999'
@8,59 SAY L32D PICTURE '99999'
@8,66 SAY L33D PICTURE '99999'
@9,2 SAY 'Jai'
@9,14 SAY L42A PICTURE '99999'
@9,21 SAY L43A PICTURE '99999'
@9,29 SAY L42B PICTURE '99999'
@9,36 SAY L43B PICTURE '99999'
@9,44 SAY L42C PICTURE '99999'
@9,51 SAY L43C PICTURE '99999'
@9,59 SAY L42D PICTURE '99999'
@9,66 SAY L43D PICTURE '99999'
@10,2 SAY 'Makhal'
@10,14 SAY L52A PICTURE '99999'
```

```
@10,21 SAY L53A PICTURE '99999'
@10,29 SAY L52B PICTURE '99999'
@10,36 SAY L53B PICTURE '99999'
@10,44 SAY L52C PICTURE '99999'
@10,51 SAY L53C PICTURE '99999'
@10,59 SAY L52D PICTURE '99999'
@10,66 SAY L53D PICTURE '99999'
@11,2 SAY 'Orah'
@11,14 SAY L62A PICTURE '99999'
@11,21 SAY L63A PICTURE '99999'
@11,29 SAY L62B PICTURE '99999'
@11,36 SAY L63B PICTURE '99999'
@11,44 SAY L62C PICTURE '99999'
@11,51 SAY L63C PICTURE '99999'
@11,59 SAY L62D PICTURE '99999'
@11,66 SAY L63D PICTURE '99999'
@12,2 SAY 'Kaliseri'
@12,14 SAY L72A PICTURE '99999'
@12,21 SAY L73A PICTURE '99999'
@12,29 SAY L72B PICTURE '99999'
@12,36 SAY L73B PICTURE '99999'
@12,44 SAY L72C PICTURE '99999'
@12,51 SAY L73C PICTURE '99999'
@12,59 SAY L72D PICTURE '99999'
@12,66 SAY L73D PICTURE '99999'
@13,2 SAY 'Tarala'
@13,14 SAY L82A PICTURE '99999'
@13,21 SAY L83A PICTURE '99999'
@13,29 SAY L82B PICTURE '99999'
@13,36 SAY L83B PICTURE '99999'
@13,44 SAY L82C PICTURE '99999'
@13,51 SAY L83C PICTURE '99999'
@13,59 SAY L82D PICTURE '99999'
@13,66 SAY L82D PICTURE '99999'
@14,2 SAY 'Barak'
@14,14 SAY L92A PICTURE '99999'
@14,21 SAY L93A PICTURE '99999'
@14,29 SAY L92B PICTURE '99999'
@14,36 SAY L93B PICTURE '99999'
@14,44 SAY L92C PICTURE '99999'
@14,51 SAY L93C PICTURE '99999'
@14,59 SAY L92D PICTURE '99999'
@14,66 SAY L93D PICTURE '99999'
@15,2 SAY 'Mitinga'
@15,14 SAY L102A PICTURE '99999'
@15,21 SAY L103A PICTURE '99999'
@15,29 SAY L102B PICTURE '99999'
@15,36 SAY L103B PICTURE '99999'
@15,44 SAY L102C PICTURE '99999'
@15,51 SAY L103C PICTURE '99999'
@15,59 SAY L102D PICTURE '99999'
@15,66 SAY L103D PICTURE '99999'
@16,2 SAY 'Muli'
@16,14 SAY L112A PICTURE '99999'
@16,21 SAY L113A PICTURE '99999'
@16,29 SAY L112B PICTURE '99999'
@16,36 SAY L113B PICTURE '99999'
@16,44 SAY L112C PICTURE '99999'
@16,51 SAY L113C PICTURE '99999'
@16,59 SAY L112D PICTURE '99999'
@16,66 SAY L113D PICTURE '99999'
@17,2 SAY 'Others'
@17,14 SAY L122A PICTURE '99999'
@17,21 SAY L123A PICTURE '99999'
@17,29 SAY L122B PICTURE '99999'
@17,36 SAY L123B PICTURE '99999'
@17,44 SAY L122C PICTURE '99999'
@17,51 SAY L123C PICTURE '99999'
@17,59 SAY L122D PICTURE '99999'
@17,66 SAY L123D PICTURE '99999'
@18,2 SAY REPLICATE('-',70)
```

```
@19,2 SAY 'Total'
@19,13 SAY A1 PICTURE '999999'
@19,20 SAY A2 PICTURE '999999'
@19,28 SAY B1 PICTURE '999999'
@19,35 SAY B2 PICTURE '999999'
@19,43 SAY C1 PICTURE '999999'
@19,50 SAY C2 PICTURE '999999'
@19,58 SAY D1 PICTURE '999999'
@19,65 SAY D2 PICTURE '999999'
@20,2 SAY REPLICATE('-',70)
@21,2 SAY 'Per Capita'
@21,14 SAY PC1 PICTURE '99.99'
@21,21 SAY PC11 PICTURE '99.99'
@21,29 SAY PC2 PICTURE '99.99'
@21,36 SAY PC22 PICTURE '99.99'
@21,44 SAY PC3 PICTURE '99.99'
@21,51 SAY PC33 PICTURE '99.99'
@21,59 SAY PC4 PICTURE '99.99'
@21,66 SAY PC44 PICTURE '99.99'
@22,2 SAY REPLICATE(CHR(196),70)
@23,2 SAY 'mc - mature culms'
@23,30 SAY 'ic - immature culms'
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!****************************************
*!    Procedure: GRASLAND
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses: PART101.DBF
*!****************************************
PROC GRASLAND
STORE 0 TO G11,G12,G13,G14,G15,G16
STORE 0 TO G21,G22,G23,G24,G25,G26
STORE 0 TO G31,G32,G33,G34,G35,G36
STORE 0 TO G41,G42,G43,G44,G45,G46
STORE 0 TO POP1,POP2,POP3,
                                        POP4,POP5,POP6

USE PART101
DO WHILE .NOT. EOF()
     GRAS=GRASS*.0929
     IF SUBSTR(SAMPLE,1,1)='1'
          POP1=POP1+FAMILY
          IF LAND='1'
               G11=G11+GRAS
          ENDIF
          IF LAND='2'
               G21=G21+GRAS
          ENDIF
          IF LAND='3'
               G31=G31+GRAS
          ENDIF
          IF LAND='4'
               G41=G41+GRAS
          ENDIF
     ENDIF
     IF SUBSTR(SAMPLE,1,1)='2'
          POP2=POP2+FAMILY
          IF LAND='1'
               G12=G12+GRAS
          ENDIF
          IF LAND='2'
               G22=G22+GRAS
          ENDIF
          IF LAND='3'
               G32=G32+GRAS
          ENDIF
```

```
      IF LAND='4'
           G42=G42+GRAS
      ENDIF
  ENDIF
  IF SUBSTR(SAMPLE,1,1)='3'
      POP3=POP3+FAMILY
      IF LAND='1'
           G13=G13+GRAS
      ENDIF
      IF LAND='2'
           G23=G23+GRAS
      ENDIF
      IF LAND='3'
           G33=G33+GRAS
      ENDIF
      IF LAND='4'
           G43=G43+GRAS
      ENDIF
  ENDIF
  IF SUBSTR(SAMPLE,1,1)='4'
      POP4=POP4+FAMILY
      IF LAND='1'
           G14=G14+GRAS
      ENDIF
      IF LAND='2'
           G24=G24+GRAS
      ENDIF
      IF LAND='3'
           G34=G34+GRAS
      ENDIF
      IF LAND='4'
           G44=G44+GRAS
      ENDIF
  ENDIF
  IF SUBSTR(SAMPLE,1,1)='5'
      POP5=POP5+FAMILY
      IF LAND='1'
           G15=G15+GRAS
      ENDIF
      IF LAND='2'
           G25=G25+GRAS
      ENDIF
      IF LAND='3'
           G35=G35+GRAS
      ENDIF
      IF LAND='4'
           G45=G45+GRAS
      ENDIF
  ENDIF
  IF SUBSTR(SAMPLE,1,1)='6'
      POP6=POP6+FAMILY
      IF LAND='1'
           G16=G16+GRAS
      ENDIF
      IF LAND='2'
           G26=G26+GRAS
      ENDIF
      IF LAND='3'
           G36=G36+GRAS
      ENDIF
      IF LAND='4'
           G46=G46+GRAS
    ' ENDIF
  ENDIF
  SKIP
ENDDO
GS1=G11+G21+G31+G41
GS2=G12+G22+G32+G42
GS3=G13+G23+G33+G43
GS4=G14+G24+G34+G44
GS5=G15+G25+G35+G45
GS6=G16+G26+G36+G46
G1T=G11+G12+G13+G14+G15+G16
G2T=G21+G22+G23+G24+G25+G26
G3T=G31+G32+G33+G34+G35+G36
G4T=G41+G42+G43+G44+G45+G46
GT=G1T+G2T+G3T+G4T
POP=POP1+POP2+POP3+POP4+
                                     POP5+POP6

GP1=GS1/POP1
GP2=GS2/POP2
GP3=GS3/POP3
GP4=GS4/POP4
GP5=GS5/POP5
GP6=GS6/POP6
GP=GT/POP
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '               Thatch Grass area in
                                  square meters'

@3,2 SAY REPLICATE(CHR(196),70)
@4,2 SAY 'Strata          1       2       3       4
                                  5       6       Total'

@5,2 SAY REPLICATE('-',70)
@6,2 SAY 'Land (acres)'
@7,2 SAY 'Less than .5'
@7,17 SAY G11 PICTURE '99999'
@7,25 SAY G12 PICTURE '99999'
@7,33 SAY G13 PICTURE '99999'
@7,41 SAY G14 PICTURE '99999'
@7,49 SAY G15 PICTURE '99999'
@7,57 SAY G16 PICTURE '99999'
@7,67 SAY G1T PICTURE '99999'
@9,2 SAY '.5 to <2.5'
@9,17 SAY G21 PICTURE '99999'
@9,25 SAY G22 PICTURE '99999'
@9,33 SAY G23 PICTURE '99999'
@9,41 SAY G24 PICTURE '99999'
@9,49 SAY G25 PICTURE '99999'
@9,57 SAY G26 PICTURE '99999'
@9,67 SAY G2T PICTURE '99999'
@11,2 SAY '2.5 to <7.5'
@11,17 SAY G31 PICTURE '99999'
@11,25 SAY G32 PICTURE '99999'
@11,33 SAY G33 PICTURE '99999'
@11,41 SAY G34 PICTURE '99999'
@11,49 SAY G35 PICTURE '99999'
@11,57 SAY G36 PICTURE '99999'
@11,67 SAY G3T PICTURE '99999'
@13,2 SAY '7.5 & above'
@13,17 SAY G41 PICTURE '99999'
@13,25 SAY G42 PICTURE '99999'
@13,33 SAY G43 PICTURE '99999'
@13,41 SAY G44 PICTURE '99999'
@13,49 SAY G45 PICTURE '99999'
@13,57 SAY G46 PICTURE '99999'
@13,67 SAY G4T PICTURE '99999'
@14,2 SAY REPLICATE('-',70)
@15,2 SAY 'Total'
@15,17 SAY GS1 PICTURE '99999'
@15,25 SAY GS2 PICTURE '99999'
@15,33 SAY GS3 PICTURE '99999'
@15,41 SAY GS4 PICTURE '99999'
@15,49 SAY GS5 PICTURE '99999'
@15,57 SAY GS6 PICTURE '99999'
@15,66 SAY GT PICTURE '999999'
@16,2 SAY REPLICATE('-',70)
@17,2 SAY 'Per capita'
@17,18 SAY GP1 PICTURE '9.99'
@17,26 SAY GP2 PICTURE '9.99'
```

```
@17,34 SAY GP3 PICTURE '9.99'
@17,42 SAY GP4 PICTURE '9.99'
@17,50 SAY GP5 PICTURE '9.99'
@17,58 SAY GP6 PICTURE '9.99'
@17,68 SAY GP PICTURE '9.99'
@18,2 SAY REPLICATE(CHR(196),70)
IF P_P='Y'
     SET DEVICE TO SCREEN
     @0,0 SAY " "
ELSE
     DO H_SCR
ENDIF
RETURN
*!**************************************
*!   Procedure: FUELUSE
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls: H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!          : PART101.DBF
*!          : PART201.DBF
*!   Indexes: ID101.NDX
*!**************************************
PROC FUELUSE
SELECT A
USE PART101 INDEX ID101
SELECT B
USE PART201

STORE 0 TO FI1,FI2,FI3,FI4,BR1,
                        BR2,BR3,BR4,TR1,TR2,TR3,TR4
STORE 0 TO BA1,BA2,BA3,BA4,AG1,AG2,
                        AG3,AG4,CO1,CO2,CO3,CO4
STORE 0 TO CH1,CH2,CH3,CH4,OI1,OI2,
                        OI3,OI4,FM1,FM2,FM3,FM4
D=1
K=.9331
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
     FI=R1C5
     BR=R2C5
     TR=R3C5
     BA=R4C5
     AG=R5C5
     CO=R6C5
     CH=R7C5
     OI=R8C5
     IF A->LAND='1'
          FI1=FI1+FI
          BR1=BR1+BR
          TR1=TR1+TR
          BA1=BA1+BA
          AG1=AG1+AG
          CO1=CO1+CO
          CH1=CH1+CH
          OI1=OI1+OI
          FM1=FM1+A->FAMILY
     ENDIF
     IF A->LAND='2'
          FI2=FI2+FI
          BR2=BR2+BR
          TR2=TR2+TR
          BA2=BA2+BA
          AG2=AG2+AG
          CO2=CO2+CO
          CH2=CH2+CH
          OI2=OI2+OI
          FM2=FM2+A->FAMILY
     ENDIF
     IF A->LAND='3'
          FI3=FI3+FI
          BR3=BR3+BR
          TR3=TR3+TR
          BA3=BA3+BA
          AG3=AG3+AG
          CO3=CO3+CO
          CH3=CH2+CH
          OI3=OI3+OI
          FM3=FM3+A->FAMILY
     ENDIF
     IF A->LAND='4'
          FI4=FI4+FI
          BR4=BR4+BR
          TR4=TR4+TR
          BA4=BA4+BA
          AG4=AG4+AG
          CO4=CO4+CO
          CH4=CH4+CH
          OI4=OI4+OI
          FM4=FM4+A->FAMILY
     ENDIF
     SKIP
ENDDO
FM=FM1+FM2+FM3+FM4
FI=(FI1+FI2+FI3+FI4)*D*K/FM
BR=(BR1+BR2+BR3+BR4)*D*K/FM
TR=(TR1+TR2+TR3+TR4)*D*K/FM
BA=(BA1+BA2+BA3+BA4)*D*K/FM
AG=(AG1+AG2+AG3+AG4)*D*K/FM
CO=(CO1+CO2+CO3+CO4)*D*K/FM
CH=(CH1+CH2+CH3+CH4)*D*K/FM
OI=(OI1+OI2+OI3+OI4)*D/FM
FI1=(FI1*D*K)/FM1
FI2=(FI2*D*K)/FM2
FI3=(FI3*D*K)/FM3
FI4=(FI4*D*K)/FM4
BR1=(BR1*D*K)/FM1
BR2=(BR2*D*K)/FM2
BR3=(BR3*D*K)/FM3
BR4=(BR4*D*K)/FM4
TR1=(TR1*D*K)/FM1
TR2=(TR2*D*K)/FM2
TR3=(TR3*D*K)/FM3
TR4=(TR4*D*K)/FM4
BA1=(BA1*D*K)/FM1
BA2=(BA2*D*K)/FM2
BA3=(BA3*D*K)/FM3
BA4=(BA4*D*K)/FM4
AG1=(AG1*D*K)/FM1
AG2=(AG2*D*K)/FM2
AG3=(AG3*D*K)/FM3
AG4=(AG4*D*K)/FM4
CO1=(CO1*D*K)/FM1
CO2=(CO2*D*K)/FM2
CO3=(CO3*D*K)/FM3
CO4=(CO4*D*K)/FM4
CH1=(CH1*D*K)/FM1
CH2=(CH2*D*K)/FM2
CH3=(CH3*D*K)/FM3
CH4=(CH4*D*K)/FM4
OI1=OI1*D/FM1
OI2=OI2*D/FM2
OI3=OI3*D/FM3
OI4=OI4*D/FM4
CLEAR
IF P_P='Y'
     SET DEVICE TO PRIN
ENDIF
@2,2 SAY '                Fuel Consumption - All Strata'
@3,2 SAY '           ( Per capita annual
                         consumption in kilogram )'
@4,2 SAY REPLICATE(CHR(196),71)
@5,2 SAY '                    Household land
```

```
                                        holding in acres'
@6,2 SAY '           ------------------
                                  -----------------------------'
@7,2 SAY 'Fuel Type   less than .5  .5 to <2.5
                       2.5 to <7.5  7.5 & above    Total'
@8,2 SAY '--------    --------   --------
                              --------    --------    -----'
@9,2 SAY 'Firewood'
@9,17 SAY FI1 PICTURE '999.9'
@9,30 SAY FI2 PICTURE '999.9'
@9,43 SAY FI3 PICTURE '999.9'
@9,56 SAY FI4 PICTURE '999.9'
@9,68 SAY FI PICTURE '999.9'
@11,2 SAY 'Branches'
@11,17 SAY BR1 PICTURE '999.9'
@11,30 SAY BR2 PICTURE '999.9'
@11,43 SAY BR3 PICTURE '999.9'
@11,56 SAY BR4 PICTURE '999.9'
@11,68 SAY BR PICTURE '999.9'
@13,2 SAY 'Tree Waste'
@13,17 SAY TR1 PICTURE '999.9'
@13,30 SAY TR2 PICTURE '999.9'
@13,43 SAY TR3 PICTURE '999.9'
@13,56 SAY TR4 PICTURE '999.9'
@13,68 SAY TR PICTURE '999.9'
@15,2 SAY 'Bamboo'
@15,17 SAY BA1 PICTURE '999.9'
@15,30 SAY BA2 PICTURE '999.9'
@15,43 SAY BA3 PICTURE '999.9'
@15,56 SAY BA4 PICTURE '999.9'
@15,68 SAY BA PICTURE '999.9'
@17,2 SAY 'Agri Residue'
@17,17 SAY AG1 PICTURE '999.9'
@17,30 SAY AG2 PICTURE '999.9'
@17,43 SAY AG3 PICTURE '999.9'
@17,56 SAY AG4 PICTURE '999.9'
@17,68 SAY AG PICTURE '999.9'
@19,2 SAY 'Cow dung'
@19,17 SAY CO1 PICTURE '999.9'
@19,30 SAY CO2 PICTURE '999.9'
@19,43 SAY CO3 PICTURE '999.9'
@19,56 SAY CO4 PICTURE '999.9'
@19,68 SAY CO PICTURE '999.9'
@21,2 SAY 'Charcoal'
@21,17 SAY CH1 PICTURE '999.9'
@21,30 SAY CH2 PICTURE '999.9'
@21,43 SAY CH3 PICTURE '999.9'
@21,56 SAY CH4 PICTURE '999.9'
@21,68 SAY CH PICTURE '999.9'
@23,2 SAY 'Oil (liter)'
@23,17 SAY OI1 PICTURE '999.9'
@23,30 SAY OI2 PICTURE '999.9'
@23,43 SAY OI3 PICTURE '999.9'
@23,56 SAY OI4 PICTURE '999.9'
@23,68 SAY OI PICTURE '999.9'
@24,2 SAY REPLICATE(CHR(196),71)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY ' '
ELSE
    DO H_SCR
ENDIF
RETURN
*!*********************************
*!    Procedure: FUELDAYS
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!         : PART101.DBF
*!         : PART201.DBF
*!    Indexes: ID101.NDX
```

```
*!*************************************
PROC FUELDAYS
SELECT A
USE PART101 INDEX ID101
SELECT B
STORE 0 TO OW1,OW2,OW3,OW4,OW5,OW6
STORE 0 TO FM1,FM2,FM3,FM4,FM5,FM6
STORE 0 TO PS1,PS2,PS3,PS4,PS5,PS6
STORE 0 TO OT1,OT2,OT3,OT4,OT5,OT6
STORE 0 TO DK1,DK2,DK3,DK4,DK5,DK6
USE PART201
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
    IF SUBSTR(A->SAMPLE,1,1)='1'
        IF FU_C='1'
            OW1=OW1+1
        ENDIF
        IF FU_C='2'
            FM1=FM1+1
        ENDIF
        IF FU_C='3'
            PS1=PS1+1
        ENDIF
        IF FU_C='4'
            OT1=OT1+1
        ENDIF
        IF FU_C='5'
            DK1=DK1+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='2'
        IF FU_C='1'
            OW2=OW2+1
        ENDIF
        IF FU_C='2'
            FM2=FM2+1
        ENDIF
        IF FU_C='3'
            PS2=PS2+1
        ENDIF
        IF FU_C='4'
            OT2=OT2+1
        ENDIF
        IF FU_C='5'
            DK2=DK2+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='3'
        IF FU_C='1'
            OW3=OW3+1
        ENDIF
        IF FU_C='2'
            FM3=FM3+1
        ENDIF
        IF FU_C='3'
            PS3=PS3+1
        ENDIF
        IF FU_C='4'
            OT3=OT3+1
        ENDIF
        IF FU_C='5'
            DK3=DK3+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='4'
        IF FU_C='1'
            OW4=OW4+1
        ENDIF
        IF FU_C='2'
            FM4=FM4+1
        ENDIF
        IF FU_C='3'
            PS4=PS4+1
        ENDIF
        IF FU_C='4'
            OT4=OT4+1
        ENDIF
        IF FU_C='5'
            DK4=DK4+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='5'
        IF FU_C='1'
            OW5=OW5+1
        ENDIF
        IF FU_C='2'
            FM5=FM5+1
        ENDIF
        IF FU_C='3'
            PS5=PS5+1
        ENDIF
        IF FU_C='4'
            OT5=OT5+1
        ENDIF
        IF FU_C='5'
            DK5=DK5+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='6'
        IF FU_C='1'
            OW6=OW6+1
        ENDIF
        IF FU_C='2'
            FM6=FM6+1
        ENDIF
        IF FU_C='3'
            PS6=PS6+1
        ENDIF
        IF FU_C='4'
            OT6=OT6+1
        ENDIF
        IF FU_C='5'
            DK6=DK6+1
        ENDIF
    ENDIF
    SKIP
ENDDO
R1=OW1+FM1+PS1+OT1+DK1
R2=OW2+FM2+PS2+OT2+DK2
R3=OW3+FM3+PS3+OT3+DK3
R4=OW4+FM4+PS4+OT4+DK4
R5=OW5+FM5+PS5+OT5+DK5
R6=OW6+FM6+PS6+OT6+DK6
POW1=(OW1/R1)*100
PFM1=(FM1/R1)*100
PPS1=(PS1/R1)*100
POT1=(OT1/R1)*100
PDK1=(DK1/R1)*100
POW2=(OW2/R2)*100
PFM2=(FM2/R2)*100
PPS2=(PS2/R2)*100
POT2=(OT2/R2)*100
PDK2=(DK2/R2)*100
POW3=(OW3/R3)*100
PFM3=(FM3/R3)*100
PPS3=(PS3/R3)*100
POT3=(OT3/R3)*100
PDK3=(DK3/R3)*100
POW4=(OW4/R4)*100
PFM4=(FM4/R4)*100
PPS4=(PS4/R4)*100
POT4=(OT4/R4)*100
PDK4=(DK4/R4)*100
POW5=(OW5/R5)*100
PFM5=(FM5/R5)*100
PPS5=(PS5/R5)*100
POT5=(OT5/R5)*100
PDK5=(DK5/R5)*100
POW6=(OW6/R6)*100
PFM6=(FM6/R6)*100
PPS6=(PS6/R6)*100
POT6=(OT6/R6)*100
PDK6=(DK6/R6)*100
RT=R1+R2+R3+R4+R5+R6
OWT=OW1+OW2+OW3+OW4+OW5+OW6
FMT=FM1+FM2+FM3+FM4+FM5+FM6
PST=PS1+PS2+PS3+PS4+PS5+PS6
OTT=OT1+OT2+OT3+OT4+OT5+OT6
DKT=DK1+DK2+DK3+DK4+DK5+DK6
POWT=(OWT/RT)*100
PFMT=(FMT/RT)*100
PPST=(PST/RT)*100
POTT=(OTT/RT)*100
PDKT=(DKT/RT)*100
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '     How many days in each week
                          is spent in fuel collection'
@3,2 SAY REPLICATE(CHR(196),69)
@4,2 SAY 'Strata          1      2      3      4
                                    5      6        Total'
@5,2 SAY REPLICATE('-',69)
@6,2 SAY 'Response no.'
@6,17 SAY R1 PICTURE '9999'
@6,25 SAY R2 PICTURE '9999'
@6,33 SAY R3 PICTURE '9999'
@6,41 SAY R4 PICTURE '9999'
@6,49 SAY R5 PICTURE '9999'
@6,57 SAY R6 PICTURE '9999'
@6,66 SAY RT PICTURE '9999'
@7,2 SAY REPLICATE('-',69)
@8,2 SAY '             %      %      %
                             %      %      %        %'
@10,2 SAY 'One day'
@10,17 SAY POW1 PICTURE '99.9'
@10,25 SAY POW2 PICTURE '99.9'
@10,33 SAY POW3 PICTURE '99.9'
@10,41 SAY POW4 PICTURE '99.9'
@10,49 SAY POW5 PICTURE '99.9'
@10,57 SAY POW6 PICTURE '99.9'
@10,66 SAY POWT PICTURE '99.9'
@12,2 SAY 'Every day'
@12,17 SAY PFM1 PICTURE '99.9'
@12,25 SAY PFM2 PICTURE '99.9'
@12,33 SAY PFM3 PICTURE '99.9'
@12,41 SAY PFM4 PICTURE '99.9'
@12,49 SAY PFM5 PICTURE '99.9'
@12,57 SAY PFM6 PICTURE '99.9'
@12,66 SAY PFMT PICTURE '99.9'
@14,2 SAY 'Two days'
@14,17 SAY PPS1 PICTURE '99.9'
@14,25 SAY PPS2 PICTURE '99.9'
@14,33 SAY PPS3 PICTURE '99.9'
@14,41 SAY PPS4 PICTURE '99.9'
@14,49 SAY PPS5 PICTURE '99.9'
@14,57 SAY PPS6 PICTURE '99.9'
@14,66 SAY PPST PICTURE '99.9'
@16,2 SAY '3 to 5 days'
@16,17 SAY POT1 PICTURE '99.9'
@16,25 SAY POT2 PICTURE '99.9'
@16,33 SAY POT3 PICTURE '99.9'
```

```
 @16,41 SAY POT4 PICTURE '99.9'
 @16,49 SAY POT5 PICTURE '99.9'
 @16,57 SAY POT6 PICTURE '99.9'
 @16,66 SAY POTT PICTURE '99.9'
 @18,2 SAY 'Not known'
 @18,17 SAY PDK1 PICTURE '99.9'
 @18,25 SAY PDK2 PICTURE '99.9'
 @18,33 SAY PDK3 PICTURE '99.9'
 @18,41 SAY PDK4 PICTURE '99.9'
 @18,49 SAY PDK5 PICTURE '99.9'
 @18,57 SAY PDK6 PICTURE '99.9'
 @18,66 SAY PDKT PICTURE '99.9'
 @19,2 SAY REPLICATE(CHR(196),69)
 IF P_P='Y'
      SET DEVICE TO SCREEN
      @0,0 SAY " "
 ELSE
      DO H_SCR
 ENDIF
 RETURN
*!**************************************
*!    Procedure: FODDFAR
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!            : PART101.DBF
*!            : PART201.DBF
*!    Indexes: ID101.NDX
*!**************************************
PROC FODDFAR
SELECT A
USE PART101 INDEX ID101
SELECT B
STORE 0 TO OW1,OW2,OW3,OW4,
                                                OW5,OW6
STORE 0 TO FM1,FM2,FM3,FM4,FM5,FM6
STORE 0 TO PS1,PS2,PS3,PS4,PS5,PS6
USE PART201
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
     IF SUBSTR(A->SAMPLE,1,1)='1'
          IF FO_F='1'
               OW1=OW1+1
          ENDIF
          IF FO_F='2'
               FM1=FM1+1
          ENDIF
          IF FO_F='3'
               PS1=PS1+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='2'
          IF FO_F='1'
               OW2=OW2+1
          ENDIF
          IF FO_F='2'
               FM2=FM2+1
          ENDIF
          IF FO_F='3'
               PS2=PS2+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='3'
          IF FO_F='1'
               OW3=OW3+1
          ENDIF
          IF FO_F='2'
               FM3=FM3+1
          ENDIF
          IF FO_F='3'
               PS3=PS3+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='4'
          IF FO_F='1'
               OW4=OW4+1
          ENDIF
          IF FO_F='2'
               FM4=FM4+1
          ENDIF
          IF FO_F='3'
               PS4=PS4+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='5'
          IF FO_F='1'
               OW5=OW5+1
          ENDIF
          IF FO_F='2'
               FM5=FM5+1
          ENDIF
          IF FO_F='3'
               PS5=PS5+1
          ENDIF
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='6'
          IF FO_F='1'
               OW6=OW6+1
          ENDIF
          IF FO_F='2'
               FM6=FM6+1
          ENDIF
          IF FO_F='3'
               PS6=PS6+1
          ENDIF
     ENDIF
     SKIP
ENDDO
R1=OW1+FM1+PS1
R2=OW2+FM2+PS2
R3=OW3+FM3+PS3
R4=OW4+FM4+PS4
R5=OW5+FM5+PS5
R6=OW6+FM6+PS6
POW1=(OW1/R1)*100
PFM1=(FM1/R1)*100
PPS1=(PS1/R1)*100
POW2=(OW2/R2)*100
PFM2=(FM2/R2)*100
PPS2=(PS2/R2)*100
POW3=(OW3/R3)*100
PFM3=(FM3/R3)*100
PPS3=(PS3/R3)*100
POW4=(OW4/R4)*100
PFM4=(FM4/R4)*100
PPS4=(PS4/R4)*100
POW5=(OW5/R5)*100
PFM5=(FM5/R5)*100
PPS5=(PS5/R5)*100
POW6=(OW6/R6)*100
PFM6=(FM6/R6)*100
PPS6=(PS6/R6)*100
RT=R1+R2+R3+R4+R5+R6
OWT=OW1+OW2+OW3+OW4+OW5+OW6
FMT=FM1+FM2+FM3+FM4+FM5+FM6
PST=PS1+PS2+PS3+PS4+PS5+PS6
POWT=(OWT/RT)*100
PFMT=(FMT/RT)*100
PPST=(PST/RT)*100
CLEAR
IF P_P='Y'
     SET DEVICE TO PRIN
```

```
ENDIF
@2,2 SAY '     How far one has to go to collect
                          fodder in rural Bangladesh'
@3,2 SAY REPLICATE(CHR(196),69)
@4,2 SAY 'Strata          1       2       3       4
                                  5       6       Total'
@5,2 SAY REPLICATE('-',69)
@6,2 SAY 'Response no.'
@6,17 SAY R1 PICTURE '9999'
@6,25 SAY R2 PICTURE '9999'
@6,33 SAY R3 PICTURE '9999'
@6,41 SAY R4 PICTURE '9999'
@6,49 SAY R5 PICTURE '9999'
@6,57 SAY R6 PICTURE '9999'
@6,66 SAY RT PICTURE '9999'
@7,2 SAY REPLICATE('-',69)
@8,2 SAY '              %       %       %
                          %       %       %       %'
@10,2 SAY '< 1 mile'
@10,18 SAY POW1 PICTURE '99.9'
@10,26 SAY POW2 PICTURE '99.9'
@10,34 SAY POW3 PICTURE '99.9'
@10,42 SAY POW4 PICTURE '99.9'
@10,50 SAY POW5 PICTURE '99.9'
@10,58 SAY POW6 PICTURE '99.9'
@10,66 SAY POWT PICTURE '99.9'
@12,2 SAY '1 - 3 miles'
@12,18 SAY PFM1 PICTURE '99.9'
@12,26 SAY PFM2 PICTURE '99.9'
@12,34 SAY PFM3 PICTURE '99.9'
@12,42 SAY PFM4 PICTURE '99.9'
@12,50 SAY PFM5 PICTURE '99.9'
@12,58 SAY PFM6 PICTURE '99.9'
@12,66 SAY PFMT PICTURE '99.9'
@14,2 SAY '> 3 miles'
@14,18 SAY PPS1 PICTURE '99.9'
@14,26 SAY PPS2 PICTURE '99.9'
@14,34 SAY PPS3 PICTURE '99.9'
@14,42 SAY PPS4 PICTURE '99.9'
@14,50 SAY PPS5 PICTURE '99.9'
@14,58 SAY PPS6 PICTURE '99.9'
@14,66 SAY PPST PICTURE '99.9'
@15,2 SAY REPLICATE(CHR(196),69)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY ""
ELSE
    DO H_SCR
ENDIF
RETURN


*!*************************************
*!    Procedure: FODDAVAI
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!          : PART101.DBF
*!          : PART201.DBF
*!    Indexes: ID101.NDX
*!*************************************
PROC FODDAVAI
SELECT A
USE PART101 INDEX ID101
SELECT B
STORE 0 TO OW1,OW2,OW3,OW4,             OW5,OW6

STORE 0 TO FM1,FM2,FM3,FM4,FM5,FM6
STORE 0 TO PS1,PS2,PS3,PS4,PS5,PS6
STORE 0 TO OT1,OT2,OT3,OT4,OT5,OT6
USE PART201
```

```
SET RELATION TO ID INTO A
DO WHILE .NOT. EOF()
    IF SUBSTR(A->SAMPLE,1,1)='1'
        IF FO_I='1'
            OW1=OW1+1
        ENDIF
        IF FO_I='2'
            FM1=FM1+1
        ENDIF
        IF FO_I='3'
            PS1=PS1+1
        ENDIF
        IF FO_I='4'
            OT1=OT1+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='2'
        IF FO_I='1'
            OW2=OW2+1
        ENDIF
        IF FO_I='2'
            FM2=FM2+1
        ENDIF
        IF FO_I='3'
            PS2=PS2+1
        ENDIF
        IF FO_I='4'
            OT2=OT2+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='3'
        IF FO_I='1'
            OW3=OW3+1
        ENDIF
        IF FO_I='2'
            FM3=FM3+1
        ENDIF
        IF FO_I='3'
            PS3=PS3+1
        ENDIF
        IF FO_I='4'
            OT3=OT3+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='4'
        IF FO_I='1'
            OW4=OW4+1
        ENDIF
        IF FO_I='2'
            FM4=FM4+1
        ENDIF
        IF FO_I='3'
            PS4=PS4+1
        ENDIF
        IF FO_I='4'
            OT4=OT4+1
        ENDIF
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='5'
        IF FO_I='1'
            OW5=OW5+1
        ENDIF
        IF FO_I='2'
            FM5=FM5+1
        ENDIF
        IF FO_I='3'
            PS5=PS5+1
        ENDIF
        IF FO_I='4'
            OT5=OT5+1
        ENDIF
        ENDIF
        IF SUBSTR(A->SAMPLE,1,1)='6'
            IF FO_I='1'
                OW6=OW6+1
            ENDIF
            IF FO_I='2'
                FM6=FM6+1
            ENDIF
            IF FO_I='3'
                PS6=PS6+1
            ENDIF
            IF FO_I='4'
                OT6=OT6+1
            ENDIF
        ENDIF
        SKIP
    ENDDO
    R1=OW1+FM1+PS1+OT1
    R2=OW2+FM2+PS2+OT2
    R3=OW3+FM3+PS3+OT3
    R4=OW4+FM4+PS4+OT4
    R5=OW5+FM5+PS5+OT5
    R6=OW6+FM6+PS6+OT6
    POW1=(OW1/R1)*100
    PFM1=(FM1/R1)*100
    PPS1=(PS1/R1)*100
    POT1=(OT1/R1)*100
    POW2=(OW2/R2)*100
    PFM2=(FM2/R2)*100
    PPS2=(PS2/R2)*100
    POT2=(OT2/R2)*100
    POW3=(OW3/R3)*100
    PFM3=(FM3/R3)*100
    PPS3=(PS3/R3)*100
    POT3=(OT3/R3)*100
    POW4=(OW4/R4)*100
    PFM4=(FM4/R4)*100
    PPS4=(PS4/R4)*100
    POT4=(OT4/R4)*100
    POW5=(OW5/R5)*100
    PFM5=(FM5/R5)*100
    PPS5=(PS5/R5)*100
    POT5=(OT5/R5)*100
    POW6=(OW6/R6)*100
    PFM6=(FM6/R6)*100
    PPS6=(PS6/R6)*100
    POT6=(OT6/R6)*100
    RT=R1+R2+R3+R4+R5+R6
    OWT=OW1+OW2+OW3+OW4+OW5+OW6
    FMT=FM1+FM2+FM3+FM4+FM5+FM6
    PST=PS1+PS2+PS3+PS4+PS5+PS6
    OTT=OT1+OT2+OT3+OT4+OT5+OT6
    POWT=(OWT/RT)*100
    PFMT=(FMT/RT)*100
    PPST=(PST/RT)*100
    POTT=(OTT/RT)*100
    CLEAR
    IF P_P='Y'
        SET DEVICE TO PRIN
    ENDIF
    @2,2 SAY '  How the repondants ranked
                        availability of fodder in their area'
    @3,2 SAY REPLICATE(CHR(196),69)
    @4,2 SAY 'Strata          1      2      3      4
                                        5      6      Total'
    @5,2 SAY REPLICATE('-',69)
    @6,2 SAY 'Response no.'
    @6,17 SAY R1 PICTURE '9999'
    @6,25 SAY R2 PICTURE '9999'
    @6,33 SAY R3 PICTURE '9999'
    @6,41 SAY R4 PICTURE '9999'
```

```
@6,49 SAY R5 PICTURE '9999'
@6,57 SAY R6 PICTURE '9999'
@6,66 SAY RT PICTURE '9999'
@7,2 SAY REPLICATE('-',69)
                            %       %       %
@8,2 SAY '
                                %       %       %       %'

@10,2 SAY 'Scanty'
@10,18 SAY POW1 PICTURE '99.9'
@10,26 SAY POW2 PICTURE '99.9'
@10,34 SAY POW3 PICTURE '99.9'
@10,42 SAY POW4 PICTURE '99.9'
@10,50 SAY POW5 PICTURE '99.9'
@10,58 SAY POW6 PICTURE '99.9'
@10,66 SAY POWT PICTURE '99.9'
@12,2 SAY 'Ample'
@12,18 SAY PFM1 PICTURE '99.9'
@12,26 SAY PFM2 PICTURE '99.9'
@12,34 SAY PFM3 PICTURE '99.9'
@12,42 SAY PFM4 PICTURE '99.9'
@12,50 SAY PFM5 PICTURE '99.9'
@12,58 SAY PFM6 PICTURE '99.9'
@12,66 SAY PFMT PICTURE '99.9'
@14,2 SAY 'Abundant'
@14,18 SAY PPS1 PICTURE '99.9'
@14,26 SAY PPS2 PICTURE '99.9'
@14,34 SAY PPS3 PICTURE '99.9'
@14,42 SAY PPS4 PICTURE '99.9'
@14,50 SAY PPS5 PICTURE '99.9'
@14,58 SAY PPS6 PICTURE '99.9'
@14,66 SAY PPST PICTURE '99.9'
@16,2 SAY 'Not known'
@16,18 SAY POT1 PICTURE '99.9'
@16,26 SAY POT2 PICTURE '99.9'
@16,34 SAY POT3 PICTURE '99.9'
@16,42 SAY POT4 PICTURE '99.9'
@16,50 SAY POT5 PICTURE '99.9'
@16,58 SAY POT6 PICTURE '99.9'
@16,66 SAY POTT PICTURE '99.9'
@17,2 SAY REPLICATE(CHR(196),69)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!**************************************
*!     Procedure: TRANSUSE
*!     Called by: PRNT (proc.in VFIPR.PRG)
*!     Calls: H_SCR (proc.in VFIPR.PRG)
*!     Uses
*!            : PDATA.DBF
*!            : PART101.DBF
*!            : PART205.DBF

*!     Indexes
*!            : PDATA.NDX
*!            : ID101.NDX
*!**************************************
PROC TRANSUSE
SELE A
USE PDATA INDEX PDATA
XX=1
DO WHILE XX<7
    X=STR(XX,1,0)
    SEEK X
    POP&X=D_PT
    XX=XX+1
ENDDO
USE PART101 INDEX ID101
SELECT B
```

```
USE PART205
SET RELATION TO ID INTO A
STORE 0 TO S1C1,S1C2,S2C1,
                         S2C2,S3C1,S3C2,S4C1,S4C2,S5C1,
                                             S5C2,S6C1,S6C2
STORE 0 TO S1C3,S2C3,S3C3,S4C3,
                                                  S5C3,S6C3
CM=.02832
DO WHILE .NOT. EOF()
    BL=BOAT_S2+BOAT_M2
                        +BOAT_L2+CART2+RICK2+
                         DULEE2+PALKI2+OTHER2
    TN=BOAT_S1*BOAT_S3+
                    BOAT_M1*BOAT_M3+BOAT_L1*
                          BOAT_L3+CART1*CART3
    TN=TN+RICK1*RICK3+
                         DULEE1*DULEE3+PALKI1*
                          PALKI3+OTHER1*OTHER3
    TL=BOAT_S1*BOAT_S4
                    +BOAT_M1*BOAT_M4+BOAT_L1
                          *BOAT_L4+CART1*CART4
    TL=TL+RICK1*RICK4+
                         DULEE1*DULEE4+PALKI1*
                          PALKI4+OTHER1*OTHER4
    IF SUBSTR(A->SAMPLE,1,1)='1'
        S1C1=S1C1+TN
        S1C2=S1C2+TL
        S1C3=S1C3+BL
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='2'
        S2C1=S2C1+TN
        S2C2=S2C2+TL
        S2C3=S2C3+BL
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='3'
        S3C1=S3C1+TN
        S3C2=S3C2+TL
        S3C3=S3C3+BL
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='4'
        S4C1=S4C1+TN
        S4C2=S4C2+TL
        S4C3=S4C3+BL
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='5'
        S5C1=S5C1+TN
        S5C2=S5C2+TL
        S5C3=S5C3+BL
    ENDIF
    IF SUBSTR(A->SAMPLE,1,1)='6'
        S6C1=S6C1+TN
        S6C2=S6C2+TL
        S6C3=S6C3+BL
    ENDIF
    SKIP
ENDDO
S1C1=S1C1*CM
S2C1=S2C1*CM
S3C1=S3C1*CM
S4C1=S4C1*CM
S5C1=S5C1*CM
S6C1=S6C1*CM
S1C2=S1C2*CM
S2C2=S2C2*CM
S3C2=S3C2*CM
S4C2=S4C2*CM
S5C2=S5C2*CM
S6C2=S6C2*CM
TOT=S1C1+S2C1+S3C1+S4C1+S5C1+S6C1
TLM=S1C2+S2C2+S3C2+S4C2+S5C2+S6C2
BTOT=S1C3+S2C3+S3C3+S4C3+
                                       S5C3+S6C3
POP=POP1+POP2+POP3+POP4+
                                       POP5+POP6
PT1=S1C1/POP1
PT2=S2C1/POP2
PT3=S3C1/POP3
PT4=S4C1/POP4
PT5=S5C1/POP5
PT6=S6C1/POP6
PL1=S1C2/POP1
PL2=S2C2/POP2
PL3=S3C2/POP3
PL4=S4C2/POP4
PL5=S5C2/POP5
PL6=S6C2/POP6
PTOT=TOT/POP
PTLM=TLM/POP
PB1=S1C3/POP1
PB2=S2C3/POP2
PB3=S3C3/POP3
PB4=S4C3/POP4
PB5=S5C3/POP5
PB6=S6C3/POP6
PBTOT=BTOT/POP
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '              Wood & Bamboo used
                                   in Transportation'
@3,2 SAY '                      ( in cubic
                                            meter )'
@4,2 SAY REPLICATE(CHR(196),71)
@5,2 SAY 'Strata     1      2      3      4
                                   5      6      Total'
@6,2 SAY REPLICATE('-',71)
@7,2 SAY 'Total Wood'
@8,2 SAY 'Volume'
@8,12 SAY S1C1 PICTURE '999.99'
@8,21 SAY S2C1 PICTURE '999.99'
@8,30 SAY S3C1 PICTURE '999.99'
@8,39 SAY S4C1 PICTURE '999.99'
@8,48 SAY S5C1 PICTURE '999.99'
@8,57 SAY S6C1 PICTURE '999.99'
@8,66 SAY TOT PICTURE '9999.99'
@10,2 SAY 'Per capita'
@10,14 SAY PT1 PICTURE '9.99'
@10,23 SAY PT2 PICTURE '9.99'
@10,32 SAY PT3 PICTURE '9.99'
@10,41 SAY PT4 PICTURE '9.99'
@10,50 SAY PT5 PICTURE '9.99'
@10,59 SAY PT6 PICTURE '9.99'
@10,69 SAY PTOT PICTURE '9.99'
@12,2 SAY REPLICATE('-',71)
@13,2 SAY 'Obtained'
@14,2 SAY 'in last 12'
@14,12 SAY S1C2 PICTURE '999.99'
@14,21 SAY S2C2 PICTURE '999.99'
@14,30 SAY S3C2 PICTURE '999.99'
@14,39 SAY S4C2 PICTURE '999.99'
@14,48 SAY S5C2 PICTURE '999.99'
@14,57 SAY S6C2 PICTURE '999.99'
@14,66 SAY TLM PICTURE '9999.99'
@15,2 SAY 'months'
@16,2 SAY REPLICATE(CHR(196),71)
@17,2 SAY 'Total'
@18,2 SAY 'Bamboo'
@18,12 SAY S1C3 PICTURE '999999'
@18,21 SAY S2C3 PICTURE '999999'
@18,30 SAY S3C3 PICTURE '999999'
@18,39 SAY S4C3 PICTURE '999999'
```

```
@18,48 SAY S5C3 PICTURE '999999'
@18,57 SAY S6C3 PICTURE '999999'
@18,66 SAY BTOT PICTURE '9999999'
@19,2 SAY '(rft)'
@21,2 SAY 'Per capita'
@22,2 SAY '(rft)'
@22,14 SAY PB1 PICTURE '99.9'
@22,23 SAY PB2 PICTURE '99.9'
@22,32 SAY PB3 PICTURE '99.9'
@22,41 SAY PB4 PICTURE '99.9'
@22,50 SAY PB5 PICTURE '99.9'
@22,59 SAY PB6 PICTURE '99.9'
@22,69 SAY PBTOT PICTURE '99.9'
@23,2 SAY REPLICATE(CHR(196),71)
IF P_P='Y'
     SET DEVICE TO SCREEN
     @0,0 SAY " "
ELSE
     DO H_SCR
ENDIF
RETURN
*!*************************************
*!    Procedure: BUILDUSE
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!          : PDATA.DBF
*!          : PART101.DBF
*!          : PART202.DBF
*!    Indexes
*!          : PDATA.NDX
*!          : ID101.NDX
*!*************************************
PROC BUILDUSE
SELE A
USE PDATA INDEX PDATA
XX=1
DO WHILE XX <7
     X=STR(XX,1,0)
     SEEK X
     POP&X=D_PT
     XX=XX+1
ENDDO
USE PART101 INDEX ID101
SELECT B
USE PART202
SET RELATION TO ID INTO A
STORE 0 TO
S1SW,S2SW,S3SW,S4SW,S5SW,S6SW,S1RW,S2RW,S3R
W,S4RW,S5RW,S6RW
STORE 0 TO S1BN,S2BN,S3BN,S4BN,S5BN,S6BN
CM=.02832
DO WHILE .NOT. EOF()
     SW=ROO2+CEI2+WAL2+
                         DOO1+PIL1+WIN1+KIT1+
                                   LAT1+OTH1+FEN3

     RW=ROO3+CEI3+WAL3+DOO2+
                         PIL2+WIN2+KIT2+LAT2+
                                        OTH2+FEN4
     BN=ROO4+CEI4+WAL4+
                         DOO3+PIL3+WIN3+KIT3+
                                   LAT3+OTH3+FEN5
     IF SUBSTR(A->SAMPLE,1,1)='1'
          S1SW=S1SW+SW
          S1RW=S1RW+RW
          S1BN=S1BN+BN
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='2'
          S2SW=S2SW+SW
          S2RW=S2RW+RW
          S2BN=S2BN+BN
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='3'
          S3SW=S3SW+SW
          S3RW=S3RW+RW
          S3BN=S3BN+BN
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='4'
          S4SW=S4SW+SW
          S4RW=S4RW+RW
          S4BN=S4BN+BN
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='5'
          S5SW=S5SW+SW
          S5RW=S5RW+RW
          S5BN=S5BN+BN
     ENDIF
     IF SUBSTR(A->SAMPLE,1,1)='6'
          S6SW=S6SW+SW
          S6RW=S6RW+RW
          S6BN=S6BN+BN
     ENDIF
     SKIP
ENDDO
S1SW=S1SW*CM
S2SW=S2SW*CM
S3SW=S3SW*CM
S4SW=S4SW*CM
S5SW=S5SW*CM
S6SW=S6SW*CM
S1RW=S1RW*CM
S2RW=S2RW*CM
S3RW=S3RW*CM
S4RW=S4RW*CM
S5RW=S5RW*CM
S6RW=S6RW*CM
TOTSW=S1SW+S2SW+S3SW+
                              S4SW+S5SW+S6SW
TOTRW=S1RW+S2RW+S3RW+
                              S4RW+S5RW+S6RW
TOTBN=S1BN+S2BN+S3BN+
                              S4BN+S5BN+S6BN
S1W=S1SW+S1RW
S2W=S2SW+S2RW
S3W=S3SW+S3RW
S4W=S4SW+S4RW
S5W=S5SW+S5RW
S6W=S6SW+S6RW
TOTW=S1W+S2W+S3W+
                                   S4W+S5W+S6W
POP=POP1+POP2+POP3+
                              POP4+POP5+POP6
PS1SW=S1SW/POP1
PS2SW=S2SW/POP2
PS3SW=S3SW/POP3
PS4SW=S4SW/POP4
PS5SW=S5SW/POP5
PS6SW=S6SW/POP6
PTOTSW=TOTSW/POP
PS1RW=S1RW/POP1
PS2RW=S2RW/POP2
PS3RW=S3RW/POP3
PS4RW=S4RW/POP4
PS5RW=S5RW/POP5
PS6RW=S6RW/POP6
PTOTRW=TOTRW/POP
PT1=S1W/POP1
PT2=S2W/POP2
PT3=S3W/POP3
PT4=S4W/POP4
PT5=S5W/POP5
```

```
PT6=S6W/POP6
PTOT=TOTW/POP
PB1=S1BN/POP1
PB2=S2BN/POP2
PB3=S3BN/POP3
PB4=S4BN/POP4
PB5=S5BN/POP5
PB6=S6BN/POP6
PBTOT=TOTBN/POP
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '               Wood & Bamboo used in
                                  Building & Fencing'
@3,2 SAY '                        ( in cubic
                                          meter )'
@4,2 SAY REPLICATE(CHR(196),71)
@5,2 SAY 'Strata      1       2       3       4
                                  5       6     Total'
@6,2 SAY REPLICATE('-',71)
@7,2 SAY 'Sawn wood'
@7,12 SAY S1SW PICTURE '999.99'
@7,21 SAY S2SW PICTURE '999.99'
@7,30 SAY S3SW PICTURE '999.99'
@7,39 SAY S4SW PICTURE '999.99'
@7,48 SAY S5SW PICTURE '999.99'
@7,57 SAY S6SW PICTURE '999.99'
@7,66 SAY TOTSW PICTURE '9999.99'
@8,2 SAY 'Per capita'
@8,14 SAY PS1SW PICTURE '.999'
@8,23 SAY PS2SW PICTURE '.999'
@8,32 SAY PS3SW PICTURE '.999'
@8,41 SAY PS4SW PICTURE '.999'
@8,50 SAY PS5SW PICTURE '.999'
@8,59 SAY PS6SW PICTURE '.999'
@8,69 SAY PTOTSW PICTURE '.999'
@10,2 SAY 'Round wood'
@10,12 SAY S1RW PICTURE '999.99'
@10,21 SAY S2RW PICTURE '999.99'
@10,30 SAY S3RW PICTURE '999.99'
@10,39 SAY S4RW PICTURE '999.99'
@10,48 SAY S5RW PICTURE '999.99'
@10,57 SAY S6RW PICTURE '999.99'
@10,66 SAY TOTRW PICTURE '9999.99'
@11,2 SAY 'Per capita'
@11,14 SAY PS1RW PICTURE '.999'
@11,23 SAY PS2RW PICTURE '.999'
@11,32 SAY PS3RW PICTURE '.999'
@11,41 SAY PS4RW PICTURE '.999'
@11,50 SAY PS5RW PICTURE '.999'
@11,59 SAY PS6RW PICTURE '.999'
@11,69 SAY PTOTRW PICTURE '.999'
@12,2 SAY REPLICATE('-',71)
@13,2 SAY 'Total wood'
@13,12 SAY S1W PICTURE '999.99'
@13,21 SAY S2W PICTURE '999.99'
@13,30 SAY S3W PICTURE '999.99'
@13,39 SAY S4W PICTURE '999.99'
@13,48 SAY S5W PICTURE '999.99'
@13,57 SAY S6W PICTURE '999.99'
@13,66 SAY TOTW PICTURE '9999.99'
@14,2 SAY 'Per capita'
@14,14 SAY PT1 PICTURE '.999'
@14,23 SAY PT2 PICTURE '.999'
@14,32 SAY PT3 PICTURE '.999'
@14,41 SAY PT4 PICTURE '.999'
@14,50 SAY PT5 PICTURE '.999'
@14,59 SAY PT6 PICTURE '.999'
@14,69 SAY PTOT PICTURE '.999'
@15,2 SAY REPLICATE('-',71)
```

```
@17,2 SAY 'Bamboo no.'
@17,12 SAY S1BN PICTURE '999999'
@17,21 SAY S2BN PICTURE '999999'
@17,30 SAY S3BN PICTURE '999999'
@17,39 SAY S4BN PICTURE '999999'
@17,48 SAY S5BN PICTURE '999999'
@17,57 SAY S6BN PICTURE '999999'
@17,66 SAY TOTBN PICTURE '9999999'
@19,2 SAY 'Per capita'
@19,14 SAY PB1 PICTURE '99.9'
@19,23 SAY PB2 PICTURE '99.9'
@19,32 SAY PB3 PICTURE '99.9'
@19,41 SAY PB4 PICTURE '99.9'
@19,50 SAY PB5 PICTURE '99.9'
@19,59 SAY PB6 PICTURE '99.9'
@19,69 SAY PBTOT PICTURE '99.9'
@20,2 SAY REPLICATE(CHR(196),71)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!****************************************
*!    Procedure: SALESPUR
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!         : PDATA.DBF
*!         : PART101.DBF
*!         : PART206.DBF
*!    Indexes
*!         : PDATA.NDX
*!         : ID101.NDX
*!****************************************
PROC SALESPUR
SELE A
USE PDATA INDEX PDATA
XX=1
DO WHILE XX<7
    X=STR(XX,1,0)
    SEEK X
    POP&X=D_PT
    XX=XX+1
ENDDO
USE PART101 INDEX ID101
SELECT B
USE PART206
SET RELATION TO ID INTO A
STORE 0 TO S1SW,S2SW,S3SW,S4SW,
                    S5SW,S6SW,S1RW,S2RW,S3RW,
                              S4RW,S5RW,S6RW

STORE 0 TO S1BN,S2BN,S3BN,S4BN,
                                   S5BN,S6BN

CM=.02832
KG=.9331
DO WHILE .NOT. EOF()
    RWQS=TIM_R1
    RWAS=TIM_R1*TIM_R2
    RWQP=TIM_R4
    RWAP=TIM_R4*TIM_R5
    SWQS=TIM_S1
    SWAS=TIM_S1*TIM_S2
    SWQP=TIM_S4
    SWAP=TIM_S4*TIM_S5
    FQS=FUEL1
    FAS=FUEL1*FUEL2
    FQP=FUEL4
    FAS=FUEL4*FUEL5
    BQS=BAM1
```

```
  BAS=BAM1*BAM2
  BQP=BAM4
  BAP=BAM4*BAM5
  IF SUBSTR(A->SAMPLE,1,1)='1'
      S1RWQS=S1RWQS+RWQS
      S1RWAS=S1RWAS+RWAS
      S1RWQP=S1RWQP+RWQP
      S1RWAP=S1RWAP+RWAP
      S1SWQS=S1SWQS+SWQS
      S1SWAS=S1SWAS+SWAS
      S1SWQP=S1SWQP+SWQP
      S1SWAP=S1SWAP+SWAP
      S1FQS=S1FQS+FQS
      S1FAS=S1FAS+FAS
      S1FQP=S1FQP+FQP
      S1FAP=S1FAP+FAP
      S1BQS=S1BQS+BQS
      S1BAS=S1BAS+BAS
      S1BQP=S1BQP+BQP
      S1BAP=S1BAP+BAP
  ENDIF
  IF SUBSTR(A->SAMPLE,1,1)='2'
      S2RWQS=S2RWQS+RWQS
      S2RWAS=S2RWAS+RWAS
      S2RWQP=S2RWQP+RWQP
      S2RWAP=S2RWAP+RWAP
      S2SWQS=S2SWQS+SWQS
      S2SWAS=S2SWAS+SWAS
      S2SWQP=S2SWQP+SWQP
      S2SWAP=S2SWAP+SWAP
      S2FQS=S2FQS+FQS
      S2FAS=S2FAS+FAS
      S2FQP=S2FAP+FQP
      S2FAP=S2FAP+FAP
      S2BQS=S2BQS+BQS
      S2BAS=S2BAS+BAS
      S2BQP=S2BQP+BQP
      S2BAP=S2BAP+BAP
  ENDIF
  IF SUBSTR(A->SAMPLE,1,1)='3'
      S3RWQS=S3RWQS+RWQS
      S3RWAS=S3RWAS+RWAS
      S3RWQP=S3RWQP+RWQP
      S3RWAP=S3RWAP+RWAP
      S3SWQS=S3SWQS+SWQS
      S3SWAS=S3SWAS+SWAS
      S3SWQP=S3SWQP+SWQP
      S3SWAP=S3SWAP+SWAP
      S3FQS=S3FQS+FQS
      S3FAS=S3FAS+FAS
      S3FQP=S3FQP+FQP
      S3FAP=S3FAP+FAP
      S3BQS=S3BQS+BQS
      S3BAS=S3BAS+BAS
      S3BQP=S3BQP+BQP
      S3BAP=S3BAP+BAP
  ENDIF
  IF SUBSTR(A->SAMPLE,1,1)='4'
      S4RWQS=S4RWQS+RWQS
      S4RWAS=S4RWAS+RWAS
      S4RWQP=S4RWQP+RWQP
      S4RWAP=S4RWAP+RWAP
      S4SWQS=S4SWQS+SWQS
      S4SWAS=S4SWAS+SWAS
      S4SWQP=S4SWQP+SWQP
      S4SWAP=S4SWAP+SWAP
      S4FQS=S4FQS+FQS
      S4FAS=S4FAS+FAS
      S4FQP=S4FQP+FQP
      S4FAP=S4FAP+FAP
      S4BQS=S4BQS+BQS
      S4BAS=S4BAS+BAS
      S4BQP=S4BQP+BQP
      S4BAP=S4BAP+BAP
  ENDIF
  IF SUBSTR(A->SAMPLE,1,1)='5'
      S5RWQS=S5RWQS+RWQS
      S5RWAS=S5RWAS+RWAS
      S5RWQP=S5RWQP+RWQP
      S5RWAP=S5RWAP+RWAP
      S5SWQS=S5SWQS+SWQS
      S5SWAS=S5SWAS+SWAS
      S5SWQP=S5SWQP+SWQP
      S5SWAP=S5SWAP+SWAP
      S5FQS=S5FQS+FQS
      S5FAS=S5FAS+FAS
      S5FQP=S5FQP+FQP
      S5FAP=S5FAP+FAP
      S5BQS=S5BQS+BQS
      S5BAS=S5BAS+BAS
      S5BQP=S5BQP+BQP
      S5BAP=S5BAP+BAP
  ENDIF
  IF SUBSTR(A->SAMPLE,1,1)='6'
      S6RWQS=S6RWQS+RWQS
      S6RWAS=S6RWAS+RWAS
      S6RWQP=S6RWQP+RWQP
      S6RWAP=S6RWAP+RWAP
      S6SWQS=S6SWQS+RWQS
      S6SWAS=S6SWAS+SWAS
      S6SWQP=S6SWQP+SWQP
      S6SWAP=S6SWAP+SWAP
      S6FQS=S6FQS+FQS
      S6FAS=S6FAS+FAS
      S6FQP=S6FQP+FQP
      S6FAP=S6FAP+FAP
      S6BQS=S6BQS+BQS
      S6BAS=S6BAS+BAS
      S6BQP=S6BQP+BQP
      S6BAP=S6BAP+BAP
  ENDIF
  SKIP
ENDDO
S1RWQS=S1RWQS*CM
S1RWQP=S1RWQP*CM
S1SWQS=S1SWQS*CM
S1SWQP=S1SWQP*CM
S1FQS=S1FQS*KG
S1FQP=S1FQP*KG
S2RWQS=S1RWQS*CM
S2RWQP=S2RWQP*CM
S2SWQS=S2SWQS*CM
S2SWQP=S2SWQP*CM
S2FQS=S2FQS*KG
S2FQP=S2FQP*KG
S3RWQS=S3RWQS*CM
S3RWQP=S3RWQP*CM
S3SWQS=S3SWQS*CM
S3SWQP=S3SWQP*CM
S3FQS=S3FQS*KG
S3FQP=S3FQP*KG
S4RWQS=S4RWQS*CM
S4RWQP=S4RWQP*CM
S4SWQS=S4SWQS*CM
S4SWQP=S4SWQP*CM
S4FQS=S4FQS*KG
S4FQP=S4FQP*KG
S5RWQS=S5RWQS*CM
S5RWQP=S5RWQP*CM
S5SWQS=S5SWQS*CM
S5SWQP=S5SWQP*CM
S5FQS=S5FQS*KG
```

```
S5FQP=S5FQP*KG
S6RWQS=S6RWQS*CM
S6RWQP=S6RWQP*CM
S6SWQS=S6SWQS*CM
S6SWQP=S6SWQP*CM
S6FQS=S6FQS*KG
S6FQP=S6FQP*KG
TRWQS=S1RWQS+S2RWQS+S3RWQS+S4RWQS+S
5RWQS+S6RWQS
TRWAS=S1RWAS+S2RWAS+S3RWAS+S4RWAS+S
5RWAS+S6RWAS
TRWQP=S1RWQP+S2RWQP+S3RWQP+S4RWQP+
S5RWQP+S6RWQP
TRWAP=S1RWAP+S2RWAP+S3RWAP+S4RWAP+S
5RWAP+S6RWAP
TSWQS=S1SWQS+S2SWQS+S3SWQS+S4SWQS+S5S
WQS+S6SWQS
TSWAS=S1SWAS+S2SWAS+S3SWAS+S4SWAS+S5S
WAS+S6SWAS
TSWQP=S1SWQP+S2SWQP+S3SWQP+S4SWQP+S5
SWQP+S6SWQP
TSWAP=S1SWAP+S2SWAP+S3SWAP+S4SWAP+S5S
WAP+S6SWAP
TFQS=S1FQS+S2FQS+S3FQS+S4FQS+S5FQS+S6FQS
TFAS=S1FAS+S2FAS+S3FAS+S4FAS+S5FAS+S6FAS
TFQP=S1FQP+S2FQP+S3FQP+S4FQP+S5FQP+S6F
QP
TFAP=S1FAP+S2FAP+S3FAP+S4FAP+S5FAP+S6FAP
TBQS=S1BQS+S2BQS+S3BQS+S4BQS+S5BQS+S6B
QS
TBAS=S1BAS+S2BAS+S3BAS+S4BAS+S5BAS+S6BAS
TBQP=S1BQP+S2BQP+S3BQP+S4BQP+S5BQP+S6
BQP
TBAP=S1BAP+S2BAP+S3BAP+S4BAP+S5BAP+S6B
AP
S1WS=S1SWQS+S1RWQS
S2WS=S2SWQS+S2RWQS
S3WS=S3SWQS+S3RWQS
S4WS=S4SWQS+S4RWQS
S5WS=S5SWQS+S5RWQS
S6WS=S6SWQS+S6RWQS
TWS=S1WS+S2WS+S3WS+S4WS+S5WS+S6WS
S1WP=S1SWQP+S1RWQP
S2WP=S2SWQP+S2RWQP
S3WP=S3SWQP+S3RWQP
S4WP=S4SWQP+S4RWQP
S5WP=S5SWQP+S5RWQP
S6WP=S6SWQP+S6RWQP
TWP=S1WP+S2WP+S3WP+S4WP+S5WP+S6WP
POP=POP1+POP2+POP3+POP4+POP5+POP6
PWS1=S1WS/POP1
PWS2=S2WS/POP2
PWS3=S3WS/POP3
PWS4=S4WS/POP4
PWS5=S5WS/POP5
PWS6=S6WS/POP6
PWST=TWS/POP
PWP1=S1WP/POP1
PWP2=S2WP/POP2
PWP3=S3WP/POP3
PWP4=S4WP/POP4
PWP5=S5WP/POP5
PWP6=S6WP/POP6
PWPT=TWP/POP
PFS1=S1FQS/POP1
PFS2=S2FQS/POP2
PFS3=S3FQS/POP3
PFS4=S4FQS/POP4
PFS5=S5FQS/POP5
PFS6=S6FQS/POP6
PFST=TFQS/POP
```

```
PFP1=S1FQP/POP1
PFP2=S2FQP/POP2
PFP3=S3FQP/POP3
PFP4=S4FQP/POP4
PFP5=S5FQP/POP5
PFP6=S6FQP/POP6
PFPT=TFQP/POP
PBS1=S1BQS/POP1
PBS2=S2BQS/POP2
PBS3=S3BQS/POP3
PBS4=S4BQS/POP4
PBS5=S5BQS/POP5
PBS6=S6BQS/POP6
PBST=TBQS/POP
PBP1=S1BQP/POP1
PBP2=S2BQP/POP2
PBP3=S3BQP/POP3
PBP4=S4BQP/POP4
PBP5=S5BQP/POP5
PBP6=S6BQP/POP6
PBPT=TBQP/POP
CLEAR
IF P_P='Y'
   SET DEVICE TO PRIN
ENDIF
@2,2 SAY '              Sales & Purchase of
                              Timber & Bamboo'
@3,2 SAY '                    ( in cubic
                                        meter )'
@4,2 SAY REPLICATE(CHR(196),71)
@5,2 SAY 'Strata      1       2       3       4
                              5       6       Total'
@6,2 SAY REPLICATE('-',71)
@7,2 SAY 'Timber Round:'
@8,2 SAY 'Qty Sold'
@8,12 SAY S1RWQS PICTURE '999999'
@8,21 SAY S2RWQS PICTURE '999999'
@8,30 SAY S3RWQS PICTURE '999999'
@8,39 SAY S4RWQS PICTURE '999999'
@8,48 SAY S5RWQS PICTURE '999999'
@8,57 SAY S6RWQS PICTURE '999999'
@8,66 SAY TRWQS PICTURE '999999'
@10,2 SAY 'Amount'
@10,12 SAY S1RWAS PICTURE '999999'
@10,21 SAY S2RWAS PICTURE '999999'
@10,30 SAY S3RWAS PICTURE '999999'
@10,39 SAY S4RWAS PICTURE '999999'
@10,48 SAY S5RWAS PICTURE '999999'
@10,57 SAY S6RWAS PICTURE '999999'
@10,66 SAY TRWAS PICTURE '999999'
@12,2 SAY 'Qty Pur.'
@12,12 SAY S1RWQP PICTURE '999999'
@12,21 SAY S2RWQP PICTURE '999999'
@12,30 SAY S3RWQP PICTURE '999999'
@12,39 SAY S4RWQP PICTURE '999999'
@12,48 SAY S5RWQP PICTURE '999999'
@12,57 SAY S6RWQP PICTURE '999999'
@12,66 SAY TRWQP PICTURE '999999'
@14,2 SAY 'Amount'
@14,12 SAY S1RWAP PICTURE '999999'
@14,21 SAY S2RWAP PICTURE '999999'
@14,30 SAY S3RWAP PICTURE '999999'
@14,39 SAY S4RWAP PICTURE '999999'
@14,48 SAY S5RWAP PICTURE '999999'
@14,57 SAY S6RWAP PICTURE '999999'
@14,66 SAY TRWAP PICTURE '999999'
@16,2 SAY 'Timber Sawn:'
@17,2 SAY 'Qty Sold'
@17,12 SAY S1SWQS PICTURE '999999'
@17,21 SAY S2SWQS PICTURE '999999'
@17,30 SAY S3SWQS PICTURE '999999'
```

```
 @17,39 SAY S4SWQS PICTURE '999999'
 @17,48 SAY S5SWQS PICTURE '999999'
 @11,2 SAY REPLICATE('-',71)
 @12,2 SAY 'Total'
 @12,12 SAY S1W PICTURE '999.99'
 @12,21 SAY S2W PICTURE '999.99'
 @12,30 SAY S3W PICTURE '999.99'
 @12,39 SAY S4W PICTURE '999.99'
 @12,48 SAY S5W PICTURE '999.99'
 @12,57 SAY S6W PICTURE '999.99'
 @12,66 SAY TOTW PICTURE '9999.99'
 @14,2 SAY 'Per capita'
 @14,14 SAY PT1 PICTURE '9.99'
 @14,23 SAY PT2 PICTURE '9.99'
 @14,32 SAY PT3 PICTURE '9.99'
 @14,41 SAY PT4 PICTURE '9.99'
 @14,50 SAY PT5 PICTURE '9.99'
 @14,59 SAY PT6 PICTURE '9.99'
 @14,69 SAY PTOT PICTURE '9.99'
 @15,2 SAY REPLICATE('-',71)
 @17,2 SAY 'Bamboo No.'
 @17,12 SAY S1BN PICTURE '999999'
 @17,21 SAY S2BN PICTURE '999999'
 @17,30 SAY S3BN PICTURE '999999'
 @17,39 SAY S4BN PICTURE '999999'
 @17,48 SAY S5BN PICTURE '999999'
 @17,57 SAY S6BN PICTURE '999999'
 @17,66 SAY TOTBN PICTURE '9999999'
 @19,2 SAY 'Per capita'
 @19,14 SAY PB1 PICTURE '99.9'
 @19,23 SAY PB2 PICTURE '99.9'
 @19,32 SAY PB3 PICTURE '99.9'
 @19,41 SAY PB4 PICTURE '99.9'
 @19,50 SAY PB5 PICTURE '99.9'
 @19,59 SAY PB6 PICTURE '99.9'
 @19,69 SAY PBTOT PICTURE '99.9'
 @20,2 SAY REPLICATE(CHR(196),71)
 IF P_P='Y'
      SET DEVICE TO SCREEN
      @0,0 SAY " "
 ELSE
      DO H_SCR
 ENDIF
 RETURN
*!***************************************
*!    Procedure: OCCULAND
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls: GET_REG (proc.in VFIPR.PRG)
*!           : H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!           : PDATA.DBF
*!           : PART101.DBF
*!    Indexes: PDATA.NDX
*!***************************************
PROC OCCULAND
S_S=0
DO GET_REG
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
P1=S_P1
P2=S_P2
P3=S_P3
P4=S_P4
PT=P1+P2+P3+P4
STORE 0 TO FA1,FA2,FA3,FA4,FI1,
                         FI2,FI3,FI4,DA1,DA2,DA3,DA4
```

```
STORE 0 TO CA1,CA2,CA3,CA4,LA1,LA2,
                         LA3,LA4,ST1,ST2,ST3,ST4
STORE 0 TO SE1,SE2,SE3,SE4,
                         OT1,OT2,OT3,OT4
USE PART101
SET FILTER TO SUBSTR(SAMPLE,1,1)=SST
GO TOP
DO WHILE .NOT. EOF()
      IF OCCUPA='1' .AND. LAND='1'
           FA1=FA1+1
      ENDIF
      IF OCCUPA='1' .AND. LAND='2'
           FA2=FA2+1
      ENDIF
      IF OCCUPA='1' .AND. LAND='3'
           FA3=FA3+1
      ENDIF
      IF OCCUPA='1' .AND. LAND='4'
           FA4=FA4+1
      ENDIF
      IF OCCUPA='2' .AND. LAND='1'
           FI1=FI1+1
      ENDIF
      IF OCCUPA='2' .AND. LAND='2'
           FI2=FI2+1
      ENDIF
      IF OCCUPA='2' .AND. LAND='3'
           FI3=FI3+1
      ENDIF
      IF OCCUPA='2' .AND. LAND='4'
           FI4=FI4+1
      ENDIF
      IF OCCUPA='3' .AND. LAND='1'
           DA1=DA1+1
      ENDIF
      IF OCCUPA='3' .AND. LAND='2'
           DA2=DA2+1
      ENDIF
      IF OCCUPA='3' .AND. LAND='3'
           DA3=DA3+1
      ENDIF
      IF OCCUPA='3' .AND. LAND='4'
           DA4=DA4+1
      ENDIF
      IF OCCUPA='4' .AND. LAND='1'
           CA1=CA1+1
      ENDIF
      IF OCCUPA='4' .AND. LAND='2'
           CA2=CA2+1
      ENDIF
      IF OCCUPA='4' .AND. LAND='3'
           CA3=CA3+1
      ENDIF
      IF OCCUPA='4' .AND. LAND='4'
           CA4=CA4+1
      ENDIF
      IF OCCUPA='5' .AND. LAND='1'
           LA1=LA1+1
      ENDIF
      IF OCCUPA='5' .AND. LAND='2'
           LA2=LA2+1
      ENDIF
      IF OCCUPA='5' .AND. LAND='3'
           LA3=LA3+1
      ENDIF
      IF OCCUPA='5' .AND. LAND='4'
           LA4=LA4+1
      ENDIF
      IF OCCUPA='6' .AND. LAND='1'
           ST1=ST1+1
      ENDIF
```

```
    IF OCCUPA='6' .AND. LAND='2'
        ST2=ST2+1
    ENDIF
    IF OCCUPA='6' .AND. LAND='3'
        ST3=ST3+1
    ENDIF
    IF OCCUPA='6' .AND. LAND='4'
        ST4=ST4+1
    ENDIF
    IF OCCUPA='7' .AND. LAND='1'
        SE1=SE1+1
    ENDIF
    IF OCCUPA='7' .AND. LAND='2'
        SE2=SE2+1
    ENDIF
    IF OCCUPA='7' .AND. LAND='3'
        SE3=SE3+1
    ENDIF
    IF OCCUPA='7' .AND. LAND='4'
        SE4=SE4+1
    ENDIF
    IF OCCUPA='8' .AND. LAND='1'
        OT1=OT1+1
    ENDIF
    IF OCCUPA='8' .AND. LAND='2'
        OT2=OT2+1
    ENDIF
    IF OCCUPA='8' .AND. LAND='3'
        OT3=OT3+1
    ENDIF
    IF OCCUPA='8' .AND. LAND='4'
        OT4=OT4+1
    ENDIF
    SKIP
ENDDO
FAT=FA1+FA2+FA3+FA4
FIT=FI1+FI2+FI3+FI4
DAT=DA1+DA2+DA3+DA4
CAT=CA1+CA2+CA3+CA4
LAT=LA1+LA2+LA3+LA4
STT=ST1+ST2+ST3+ST4
SERT=SE1+SE2+SE3+SE4
OTT=OT1+OT2+OT3+OT4
L1=FA1+FI1+DA1+CA1+
                        LA1+ST1+SE1+OT1
L2=FA2+FI2+DA2+CA2+
                        LA2+ST2+SE2+OT2
L3=FA3+FI3+DA3+CA3+
                        LA3+ST3+SE3+OT3
L4=FA4+FI4+DA4+CA4+
                        LA4+ST4+SE4+OT4
LT=L1+L2+L3+L4
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '   Population, Occupation and Land
                Ownership (in acres) - Stratum '+SST
@3,2 SAY REPLICATE(CHR(196),74)
@4,2 SAY '          Less than .50  .50 to <2.50
                          2.5 to <7.5   7.5 & above'
@5,2 SAY '          ----------   ----------
                               ----------   ----------'
@6,2 SAY 'Occupation        No.          No.
                             No.         No.       Total'
@7,2 SAY REPLICATE('-',74)
@8,2 SAY 'Farming'
@8,17 SAY FA1 PICTURE '9999'
@8,31 SAY FA2 PICTURE '9999'
@8,45 SAY FA3 PICTURE '9999'
@8,59 SAY FA4 PICTURE '9999'
@8,72 SAY FAT PICTURE '9999'
@9,2 SAY 'Fishing'
@9,17 SAY FI1 PICTURE '9999'
@9,31 SAY FI2 PICTURE '9999'
@9,45 SAY FI3 PICTURE '9999'
@9,59 SAY FI4 PICTURE '9999'
@9,72 SAY FIT PICTURE '9999'
@10,2 SAY 'Diary/Poultry'
@10,17 SAY DA1 PICTURE '9999'
@10,31 SAY DA2 PICTURE '9999'
@10,45 SAY DA3 PICTURE '9999'
@10,59 SAY DA4 PICTURE '9999'
@10,72 SAY DAT PICTURE '9999'
@11,2 SAY 'Crafts'
@11,17 SAY CA1 PICTURE '9999'
@11,31 SAY CA2 PICTURE '9999'
@11,45 SAY CA3 PICTURE '9999'
@11,59 SAY CA4 PICTURE '9999'
@11,72 SAY CAT PICTURE '9999'
@12,2 SAY 'Labourer'
@12,17 SAY LA1 PICTURE '9999'
@12,31 SAY LA2 PICTURE '9999'
@12,45 SAY LA3 PICTURE '9999'
@12,59 SAY LA4 PICTURE '9999'
@12,72 SAY LAT PICTURE '9999'
@13,2 SAY 'Small trade'
@13,17 SAY ST1 PICTURE '9999'
@13,31 SAY ST2 PICTURE '9999'
@13,45 SAY ST3 PICTURE '9999'
@13,59 SAY ST4 PICTURE '9999'
@13,72 SAY STT PICTURE '9999'
@14,2 SAY 'Service'
@14,17 SAY SE1 PICTURE '9999'
@14,31 SAY SE2 PICTURE '9999'
@14,45 SAY SE3 PICTURE '9999'
@14,59 SAY SE4 PICTURE '9999'
@14,72 SAY SERT PICTURE '9999'
@15,2 SAY 'Other'
@15,17 SAY OT1 PICTURE '9999'
@15,31 SAY OT2 PICTURE '9999'
@15,45 SAY OT3 PICTURE '9999'
@15,59 SAY OT4 PICTURE '9999'
@15,72 SAY OTT PICTURE '9999'
@16,2 SAY REPLICATE("-",74)
@17,2 SAY 'Total Sample'
@17,17 SAY L1 PICTURE '9999'
@17,31 SAY L2 PICTURE '9999'
@17,45 SAY L3 PICTURE '9999'
@17,59 SAY L4 PICTURE '9999'
@17,72 SAY LT PICTURE '9999'
@18,2 SAY REPLICATE('-',74)
@19,2 SAY 'Population'
@19,16 SAY P1 PICTURE '99999'
@19,30 SAY P2 PICTURE '99999'
@19,44 SAY P3 PICTURE '99999'
@19,58 SAY P4 PICTURE '99999'
@19,71 SAY PT PICTURE '99999'
@20,2 SAY REPLICATE(CHR(196),74)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*|****************************************
*|    Procedure: BAMCLUMP
*|    Called by: PRNT (proc.in VFIPR.PRG)
*|    Calls
*|          : GET_REG (proc.in VFIPR.PRG)
*|          : H_SCR (proc.in VFIPR.PRG)
*|
 *!    Uses : PDATA.DBF
 *!           : PART101.DBF
 *!           : PART102.DBF
 *!    Indexes
 *!           : PDATA.NDX
 *!           : ID101.NDX
 *!*************************************
 PROC BAMCLUMP
 S_S=0
 DO GET_REG
 IF S_S=0
     RETURN
 ENDIF
 SST=STR(S_S,1,0)
 SELE A
 USE PDATA INDEX PDATA
 SEEK SST
 POP=S_PT
 TPOP=T_POP
 USE PART101 INDEX ID101
 SELECT B
 USE PART102
 SET RELATION TO ID INTO A
 SET FILTER TO
                        SUBSTR(A->SAMPLE,1,1)=SST
 GO TOP
 STORE 0 TO L11,L21,L31,L41,
                                   L51,L61,L71,L81,
                                 L91,L101,L111,L121
 STORE 0 TO TMC,TTC
 DO WHILE .NOT. EOF()
      L11=L11+B1_CL
      L21=L21+B2_CL
      L31=L31+B3_CL
      L41=L41+B4_CL
      L51=L51+B5_CL
      L61=L61+B6_CL
      L71=L71+B7_CL
      L81=L81+B8_CL
      L91=L91+B9_CL
      L101=L101+B10_CL
      L111=L111+B11_CL
      L121=L121+B12_CL
      SKIP
ENDDO
IT=1
LN=6
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@1,2 SAY '            Bamboo Resources -
                                       Stratum '+SST
@2,2 SAY REPLICATE(CHR(196),55)
@3,2 SAY '               No. of Clumps     Per
                                                Total'
@4,2 SAY 'Species       in Sample Pop.
                                   Capita     Clumps'
@5,2 SAY '-------          -------      ------
                                                ------'
DO WHILE IT<=12
    IF IT=1
        NAM='Katabash'
        MC=L11
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=2
        NAM='Bariala'
        MC=L21
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=3
        NAM='Barua'
        MC=L31
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=4
        NAM='Jai'
        MC=L41
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=5
        NAM='Makhal'
        MC=L51
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=6
        NAM='Orah'
        MC=L61
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=7
        NAM='Kaliseri'
        MC=L71
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=8
        NAM='Tarala'
        MC=L81
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=9
        NAM='Barak'
        MC=L91
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=10
        NAM='Mitinga'
        MC=L101
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=11
        NAM='Muli'
        MC=L111
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    IF IT=12
        NAM='Others'
        MC=L121
        PMC=MC/POP
        TC=PMC*TPOP
    ENDIF
    TMC=TMC+MC
    TTC=TTC+TC
    @LN,2 SAY NAM
    @LN,24 SAY MC PICTURE '99999'
    @LN,37 SAY PMC PICTURE '.9999'
    @LN,48 SAY TC PICTURE '99999999'
    LN=LN+1
    IT=IT+1
ENDDO
```

```
@LN,2 SAY REPLICATE('-',55)
@LN+1,9 SAY 'Total:'
@LN+1,23 SAY TMC PICTURE '999999'
@LN+1,48 SAY TTC PICTURE '99999999'
@LN+2,2 SAY REPLICATE(CHR(196),55)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!**************************************
*!   Procedure: STOCPALM
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls
*!        : GET_REG (proc.in VFIPR.PRG)
*!        : H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!        : PDATA.DBF
*!        : PART101.DBF
*!        : PART105.DBF
*!   Indexes
*!        : PDATA.NDX
*!        : ID101.NDX
*!**************************************


PROC STOCPALM
S_S=0
DO GET_REG
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
POP=S_PT
USE PART101 INDEX ID101
SELECT B
STORE 0 TO T1,T2,T3,N1,N2,
                          N3,K1,K2,K3,S1,S2,S3
STORE 0 TO L1,L2,L3,LT,LPOP
USE PART105
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@1,2 SAY '          Stock of Palm trees -
                                    Stratum '+SST
@2,2 SAY REPLICATE(CHR(196),53)
@3,2 SAY '          Number according to height
                                         in ft.'
@4,2 SAY 'Local       ---------------------------'
@5,2 SAY 'Name     5 - 10    10 - 20
                           Above 20    Total'
@6,2 SAY '---      -----    -----    -----    ----'

DO WHILE .NOT. EOF()
    T1=T1+TAL1
    T2=T2+TAL2
    T3=T3+TAL3
    N1=N1+NAR1
    N2=N2+NAR2
    N3=N3+NAR3
    K1=K1+KHE1
    K2=K2+KHE2
    K3=K3+KHE3
    S1=S1+SUP1
    S2=S2+SUP2
    S3=S3+SUP3
    SKIP
ENDDO
T=T1+T2+T3
N=N1+N2+N3
K=K1+K2+K3
S=S1+S2+S3
L1=T1+N1+K1+S1
L2=T2+N2+K2+S2
L3=T3+N3+K3+S3
LT=T+N+K+S
PT=T/POP
PN=N/POP
PK=K/POP
PS=S/POP
LPOP=LT/POP
@7,2 SAY 'Tal'
@7,15 SAY T1 PICTURE '99999'
@7,27 SAY T2 PICTURE '99999'
@7,39 SAY T3 PICTURE '99999'
@7,49 SAY T PICTURE '999999'
@8,2 SAY 'Per capita'
@8,16 SAY T1/POP PICTURE '.999'
@8,28 SAY T2/POP PICTURE '.999'
@8,40 SAY T3/POP PICTURE '.999'
@8,51 SAY PT PICTURE '.999'
@10,2 SAY 'Narkel'
@10,15 SAY N1 PICTURE '99999'
@10,27 SAY N2 PICTURE '99999'
@10,39 SAY N3 PICTURE '99999'
@10,49 SAY N PICTURE '999999'
@11,2 SAY 'Per capita'
@11,16 SAY N1/POP PICTURE '.999'
@11,28 SAY N2/POP PICTURE '.999'
@11,40 SAY N3/POP PICTURE '.999'
@11,50 SAY PN PICTURE '9.999'
@13,2 SAY 'Khejur'
@13,15 SAY K1 PICTURE '99999'
@13,27 SAY K2 PICTURE '99999'
@13,39 SAY K3 PICTURE '99999'
@13,49 SAY K PICTURE '999999'
@14,2 SAY 'Per capita'
@14,16 SAY K1/POP PICTURE '.999'
@14,28 SAY K2/POP PICTURE '.999'
@14,40 SAY K3/POP PICTURE '.999'
@14,50 SAY PK PICTURE '9.999'
@16,2 SAY 'Supari'
@16,15 SAY S1 PICTURE '99999'
@16,27 SAY S2 PICTURE '99999'
@16,39 SAY S3 PICTURE '99999'
@16,49 SAY S PICTURE '999999'
@17,2 SAY 'Per capita'
@17,15 SAY S1/POP PICTURE '9.999'
@17,27 SAY S2/POP PICTURE '9.999'
@17,39 SAY S3/POP PICTURE '9.999'
@17,50 SAY PS PICTURE '99.99'
@18,2 SAY REPLICATE(CHR(196),53)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!**************************************
*!   Procedure: SMALLAND
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls
*!         : GET_REG (proc.in VFIPR.PRG)
*!         : H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!         : PDATA.DBF
*!         : PART101.DBF
*!         : PART1031.DBF
*!   Indexes
*!         : PDATA.NDX
*!         : ID101.NDX
*!         : CODE1031.NDX
*!************************************
PROC SMALLAND
S_S=0
DO GET_REG
IF S_S=0
     RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
SP=T_POP
POP1=S_P1
POP2=S_P2
POP3=S_P3
POP4=S_P4
R1=P_R1
R2=P_R2
R3=P_R3
R4=P_R4
USE PART101 INDEX ID101
SELECT B
USE PART1031 INDEX CODE1031
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP
STORE 0 TO TOT1,TOT2,TOT3,TOT4
LN=5
CLEAR
IF P_P='Y'
     SET DEVICE TO PRIN
ENDIF
@1,2 SAY '    Stock of trees upto 8" dia(BH)
                               & greater than 5ft height'
@2,13 SAY 'in terms of land owned (in acres)
                                        - Stratum '+SST
@3,2 SAY REPLICATE(CHR(196),69)
@4,3 SAY 'Species    less than .5 .5 to <2.5
                        2.5 to <7.5  7.5 & above    All'
@5,3 SAY '------   ------------ --------- -----------
                                     ----------- ------'
DO WHILE .NOT. EOF()
     STORE 0 TO N1,N2,N3,N4
     LN=LN+1
     CD=CODE
     DO WHILE CODE=CD
          IF A->LAND='1'
               N1=N1+NUMBER
          ENDIF
          IF A->LAND='2'
               N2=N2+NUMBER
          ENDIF
          IF A->LAND='3'
               N3=N3+NUMBER
          ENDIF
          IF A->LAND='4'
               N4=N4+NUMBER
          ENDIF
          SKIP
     ENDDO
     IF CD='01'
          X='Mango'
     ENDIF
     IF CD='02'
          X='Jack'
     ENDIF
     IF CD='03'
          X='Rain'
     ENDIF
     IF CD='04'
          X='Simul'
     ENDIF
     IF CD='05'
          X='Bat'
     ENDIF
     IF CD='06'
          X='Madar'
     ENDIF
     IF CD='07'
          X='Koroi'
     ENDIF
     IF CD='08'
          X='Chakua'
     ENDIF
     IF CD='09'
          X='Jam'
     ENDIF
     IF CD='10'
          X='Jiul'
     ENDIF
     IF CD='11'
          X='Gab'
     ENDIF
     IF CD='12'
          X='Tetul'
     ENDIF
     IF CD='13'
          X='Bel'
     ENDIF
     IF CD='14'
          X='Pitali'
     ENDIF
     IF CD='15'
          X='Chhatim'
     ENDIF
     IF CD='16'
          X='Kadam'
     ENDIF
     IF CD='17'
          X='Debdaru'
     ENDIF
     IF CD='18'
          X='Jarul'
     ENDIF
     IF CD='19'
          X='Sal'
     ENDIF
     IF CD='20'
          X='Segun'
     ENDIF
     IF CD='21'
          X='Garjan'
     ENDIF
     IF CD='22'
          X='Palash'
     ENDIF
     IF CD='23'
          X='Lichu'
     ENDIF
     IF CD='24'
          X='Others'
     ENDIF
```

```
        N1=(N1/POP1)*SP*R1/1000
        N2=(N2/POP2)*SP*R2/1000
        N3=(N3/POP3)*SP*R3/1000
        N4=(N4/POP4)*SP*R4/1000
        NT=N1+N2+N3+N4
        @LN,3 SAY X
        @LN,16 SAY N1 PICTURE '999999'
        @LN,29 SAY N2 PICTURE '999999'
        @LN,41 SAY N3 PICTURE '999999'
        @LN,54 SAY N4 PICTURE '999999'
        @LN,65 SAY NT PICTURE '999999'
        TOT1=TOT1+N1
        TOT2=TOT2+N2
        TOT3=TOT3+N3
        TOT4=TOT4+N4
  ENDDO
  TOTT=TOT1+TOT2+TOT3+TOT4
  @LN+1,2 SAY REPLICATE('-',69)
  @LN+2,2 SAY 'All Species'
  @LN+2,16 SAY TOT1 PICTURE '999999'
  @LN+2,29 SAY TOT2 PICTURE '999999'
  @LN+2,41 SAY TOT3 PICTURE '999999'
  @LN+2,54 SAY TOT4 PICTURE '999999'
  @LN+2,65 SAY TOTT PICTURE '999999'
  @LN+3,2 SAY REPLICATE(CHR(196),69)
  @LN+5,2 SAY '* All figures are in 000'
  IF P_P='Y'
      SET DEVICE TO SCREEN
      @0,0 SAY " "
  ELSE
      DO H_SCR
  ENDIF
  RETURN
*!**************************************
*!   Procedure: STOCMID
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls: H_SCR (proc.in VFIPR.PRG)
*!   Uses
*!          : PDATA.DBF
*!          : PART104.DBF
*!   Indexes
*!          : PDATA.NDX
*!          : CODE104.NDX
*!**************************************
PROC STOCMID
SELE A
USE PDATA INDEX PDATA
SEEK 'T'
TP=S_PT
TOT=0
LN=5
USE PART104 INDEX CODE104
SET FILTER TO GIRTH <=25
GO TOP
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@1,2 SAY '     Stock of trees of dia(BH)
*!          between 4" and 8" '
@2,2 SAY '                    ( All Strata )'
@3,2 SAY REPLICATE(CHR(196),57)
@4,7 SAY 'ID       Species      Stems
                                        Per Capita'

@5,7 SAY '--       -------      -----
                                        ----------'

DO WHILE .NOT. EOF()
    CD=CODE
    NO=0
    LN=LN+1
    DO WHILE CODE=CD
```

```
        NO=NO+1
        SKIP
ENDDO
PC=NO/TP
TOT=TOT+NO
IF CD='01'
    X='Mango'
ENDIF
IF CD='02'
    X='Jack'
ENDIF
IF CD='03'
    X='Rain'
ENDIF
IF CD='04'
    X='Simul'
ENDIF
IF CD='05'
    X='Bat'
ENDIF
IF CD='06'
    X='Madar'
ENDIF
IF CD='07'
    X='Koroi'
ENDIF
IF CD='08'
    X='Chakua'
ENDIF
IF CD='09'
    X='Jam'
ENDIF
IF CD='10'
    X='Jiul'
ENDIF
IF CD='11'
    X='Gab'
ENDIF
IF CD='12'
    X='Tetul'
ENDIF
IF CD='13'
    X='Bel'
ENDIF
IF CD='14'
    X='Pitali'
ENDIF
IF CD='15'
    X='Chhatim'
ENDIF
IF CD='16'
    X='Kadam'
ENDIF
IF CD='17'
    X='Debdaru'
ENDIF
IF CD='18'
    X='Jarul'
ENDIF
IF CD='19'
    X='Sal'
ENDIF
IF CD='20'
    X='Segun'
ENDIF
IF CD='21'
    X='Garjan'
ENDIF
IF CD='22'
    X='Palash'
ENDIF
```

```
     IF CD='23'
          X='Lichu'
     ENDIF
     IF CD='24'
          X='Others'
     ENDIF
     @LN,7 SAY CD
     @LN,17 SAY X
     @LN,31 SAY NO PICTURE '99999'
     @LN,47 SAY PC PICTURE '99.99'
ENDDO
PC=TOT/TP
LN=LN+1
@LN,7 SAY REPLICATE('-',45)
@LN+1,17 SAY 'All species:'
@LN+1,31 SAY TOT PICTURE '99999'
@LN+1,47 SAY PC PICTURE '99.99'
@LN+2,2 SAY REPLICATE(CHR(196),57)
IF P_P='Y'
     SET DEVICE TO SCREEN
     @0,0 SAY " "
ELSE
     DO H_SCR
ENDIF
RETURN

*!***************************************
*!   Procedure: POPULAND
*!   Called by: PRNT (proc.in VFIPR.PRG)
*!   Calls: H_SCR (proc.in VFIPR.PRG)
*!   Uses: PART101.DBF
*!***************************************
PROC POPULAND
SET TALK OFF
STORE 0 TO S1L1,S1L2,S1L3,S1L4,
                                   S2L1,S2L2,S2L3,S2L4
STORE 0 TO S3L1,S3L2,S3L3,S3L4,
                                   S4L1,S4L2,S4L3,S4L4
STORE 0 TO S5L1,S5L2,S5L3,S5L4,
                                   S6L1,S6L2,S6L3,S6L4
USE PART101
DO WHILE .NOT. EOF()
     IF SUBSTR(SAMPLE,1,1)='1'
          IF LAND='1'
               S1L1=S1L1+FAMILY
          ENDIF
          IF LAND='2'
               S1L2=S1L2+FAMILY
          ENDIF
          IF LAND='3'
               S1L3=S1L3+FAMILY
          ENDIF
          IF LAND='4'
               S1L4=S1L4+FAMILY
          ENDIF
     ENDIF
     IF SUBSTR(SAMPLE,1,1)='2'
          IF LAND='1'
               S2L1=S2L1+FAMILY
          ENDIF
          IF LAND='2'
               S2L2=S2L2+FAMILY
          ENDIF
          IF LAND='3'
               S2L3=S2L3+FAMILY
          ENDIF
          IF LAND='4'
               S2L4=S2L4+FAMILY
          ENDIF
     ENDIF
     IF SUBSTR(SAMPLE,1,1)='3'
          IF LAND='1'
               S3L1=S3L1+FAMILY
          ENDIF
          IF LAND='2'
               S3L2=S3L2+FAMILY
          ENDIF
          IF LAND='3'
               S3L3=S3L3+FAMILY
          ENDIF
          IF LAND='4'
               S3L4=S3L4+FAMILY
          ENDIF
          IF LAND='5'
               S3L5=S3L5+FAMILY
          ENDIF
     ENDIF
     IF SUBSTR(SAMPLE,1,1)='4'
          IF LAND='1'
               S4L1=S4L1+FAMILY
          ENDIF
          IF LAND='2'
               S4L2=S4L2+FAMILY
          ENDIF
          IF LAND='3'
               S4L3=S4L3+FAMILY
          ENDIF
          IF LAND='4'
               S4L4=S4L4+FAMILY
          ENDIF
     ENDIF
     IF SUBSTR(SAMPLE,1,1)='5'
          IF LAND='1'
               S5L1=S5L1+FAMILY
          ENDIF
          IF LAND='2'
               S5L2=S5L2+FAMILY
          ENDIF
          IF LAND='3'
               S5L3=S5L3+FAMILY
          ENDIF
          IF LAND='4'
               S5L4=S5L4+FAMILY
          ENDIF
     ENDIF
     IF SUBSTR(SAMPLE,1,1)='6'
          IF LAND='1'
               S6L1=S6L1+FAMILY
          ENDIF
          IF LAND='2'
               S6L2=S6L2+FAMILY
          ENDIF
          IF LAND='3'
               S6L3=S6L3+FAMILY
          ENDIF
          IF LAND='4'
               S6L4=S6L4+FAMILY
          ENDIF
     ENDIF
     SKIP
ENDDO
S1=S1L1+S1L2+S1L3+S1L4
S2=S2L1+S2L2+S2L3+S2L4
S3=S3L1+S3L2+S3L3+S3L4
S4=S4L1+S4L2+S4L3+S4L4
S5=S5L1+S5L2+S5L3+S5L4
S6=S6L1+S6L2+S6L3+S6L4
L1=S1L1+S2L1+S3L1+S4L1+S5L1+S6L1
L2=S1L2+S2L2+S3L2+S4L2+S5L2+S6L2
L3=S1L3+S2L3+S3L3+S4L3+S5L3+S6L3
L4=S1L4+S2L4+S3L4+S4L4+S5L4+S6L4
LT=L1+L2+L3+L4
```

```
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@2,2 SAY '                    Population
                                          Distribution  '
@3,2 SAY REPLICATE(CHR(196),73)
@4,2 SAY '          Less than .50  .50 to <2.50
                               2.5 to <7.5  7.5 & above'
@5,2 SAY '          ----------  ----------
                                    ----------  ----------'
@6,2 SAY 'Stratum         No.          No.
                           No.          No.      Total'
@7,2 SAY REPLICATE('-',73)
@8,2 SAY 'Stratum 1'
@8,17 SAY S1L1 PICTURE '99999'
@8,31 SAY S1L2 PICTURE '99999'
@8,45 SAY S1L3 PICTURE '99999'
@8,59 SAY S1L4 PICTURE '99999'
@8,70 SAY S1 PICTURE '99999'
@9,2 SAY 'Stratum 2'
@9,17 SAY S2L1 PICTURE '99999'
@9,31 SAY S2L2 PICTURE '99999'
@9,45 SAY S2L3 PICTURE '99999'
@9,59 SAY S2L4 PICTURE '99999'
@9,70 SAY S2 PICTURE '99999'
@10,2 SAY 'Stratum 3'
@10,17 SAY S3L1 PICTURE '99999'
@10,31 SAY S3L2 PICTURE '99999'
@10,45 SAY S3L3 PICTURE '99999'
@10,59 SAY S3L4 PICTURE '99999'
@10,70 SAY S3 PICTURE '99999'
@11,2 SAY 'Stratum 4'
@11,17 SAY S4L1 PICTURE '99999'
@11,31 SAY S4L2 PICTURE '99999'
@11,45 SAY S4L3 PICTURE '99999'
@11,59 SAY S4L4 PICTURE '99999'
@11,70 SAY S4 PICTURE '99999'
@12,2 SAY 'Stratum 5'
@12,17 SAY S5L1 PICTURE '99999'
@12,31 SAY S5L2 PICTURE '99999'
@12,45 SAY S5L3 PICTURE '99999'
@12,59 SAY S5L4 PICTURE '99999'
@12,70 SAY S5 PICTURE '99999'
@13,2 SAY 'Stratum 6'
@13,17 SAY S6L1 PICTURE '99999'
@13,31 SAY S6L2 PICTURE '99999'
@13,45 SAY S6L3 PICTURE '99999'
@13,59 SAY S6L4 PICTURE '99999'
@13,70 SAY S6 PICTURE '99999'
@14,2 SAY REPLICATE("-",73)
@15,2 SAY 'Total'
@15,17 SAY L1 PICTURE '99999'
@15,31 SAY L2 PICTURE '99999'
@15,45 SAY L3 PICTURE '99999'
@15,59 SAY L4 PICTURE '99999'
@15,70 SAY LT PICTURE '99999'
@16,2 SAY REPLICATE(CHR(196),73)
IF P_P='Y'
    SET DEVICE TO SCREEN
    @0,0 SAY " "
ELSE
    DO H_SCR
ENDIF
RETURN
*!**************************************
*!    Procedure: TREELAND
*!    Called by: PRNT (proc.in VFIPR.PRG)
*!    Calls
*!         : GET_REG (proc.in VFIPR.PRG)
*!         : H_SCR (proc.in VFIPR.PRG)
*!    Uses
*!         : PDATA.DBF
*!         : PART101.DBF
*!         : PART104.DBF
*!    Indexes
*!         : PDATA.NDX
*!         : ID101.NDX
*!         : CODE104.NDX
*!**************************************
PROC TREELAND
S_S=0
DO GET_REG
IF S_S=0
    RETURN
ENDIF
SST=STR(S_S,1,0)
SELE A
USE PDATA INDEX PDATA
SEEK SST
SP=T_POP
POP1=S_P1
POP2=S_P2
POP3=S_P3
POP4=S_P4
POP=S_PT
R1=P_R1
R2=P_R2
R3=P_R3
R4=P_R4
USE PART101 INDEX ID101
SELECT B
USE PART104 INDEX CODE104
SET RELATION TO ID INTO A
SET FILTER TO SUBSTR(A->SAMPLE,1,1)=SST
GO TOP


STORE 0 TO TS1,TS2,TS3,TS4,TVF1,
                             TVF2,TVF3,TVF4,
                         TVS1,TVS2,TVS3,TVS4
STORE 0 TO POP1,POP2,POP3,POP4
CM=.02832
CLEAR
IF P_P='Y'
    SET DEVICE TO PRIN
ENDIF
@1,2 SAY '          Stock Volume by Land
                         Holding - Stratum '+SST
@2,2 SAY REPLICATE(CHR(196),63)
@3,2 SAY 'Land holding   Total vol   Per
                   Capita   Sawlog vol   Fire wood'
@4,2 SAY '(acres)        (000 cm)     (cu m)
                            (000 cm) (f)   (000 cm)'
@5,2 SAY REPLICATE('-',63)
DO WHILE CODE='01'
    DBH=GIRTH/3.14159
    IF BOLE_HT>0
        DBOT=((DBH*BOLE_HT*12)-
                     (8*54))/(BOLE_HT*12-54)
        DMID=(DBOT+8)/2
    ELSE
        DBOT=0
        DMID=0
    ENDIF
    BT2=.54+(.039*DMID)
    VF=(-11.0739+(.2576*DBH^2))
    RUB=(DMID-BT2)/2
    VS=3.14159*(RUB/12)^2*BOLE_HT
    IF LAND='1'
        TS1=TS1+1
        TVF1=TVF1+VF
```

```
          TVS1=TVS1+VS
     ENDIF
     IF LAND='2'
          TS2=TS2+1
          TVF2=TVF2+VF
          TVS2=TVS2+VS
     ENDIF
     IF LAND='3'
          TS3=TS3+1
          TVF3=TVF3+VF
          TVS3=TVS3+VS
     ENDIF
     IF LAND='4'
          TS4=TS4+1
          TVF4=TVF4+VF
          TVS4=TVS4+VS
     ENDIF
     SKIP
ENDDO
DO WHILE CODE='02'
     DBH=GIRTH/3.14159
     IF BOLE_HT>0
          DBOT=((DBH*BOLE_HT*12)
                              -(8*54))/(BOLE_HT*12-54)
          DMID=(DBOT+8)/2
     ELSE
          DBOT=0
          DMID=0
     ENDIF
     BT2=.79+(.025*DMID)
     VF=(-6.9127+(.1824*DBH^2))
     RUB=(DMID-BT2)/2
     VS=3.14159*(RUB/12)^2*BOLE_HT
     IF LAND='1'
          TS1=TS1+1
          TVF1=TVF1+VF
          TVS1=TVS1+VS
     ENDIF
     IF LAND='2'
          TS2=TS2+1
          TVF2=TVF2+VF
          TVS2=TVS2+VS
     ENDIF
     IF LAND='3'
          TS3=TS3+1
          TVF3=TVF3+VF
          TVS3=TVS3+VS
     ENDIF
     IF LAND='4'
          TS4=TS4+1
          TVF4=TVF4+VF
          TVS4=TVS4+VS
     ENDIF
     SKIP
ENDDO
DO WHILE .NOT. EOF()
     DBH=GIRTH/3.14159
     IF BOLE_HT>0
          DBOT=((DBH*BOLE_HT*12)-
                              (8*54))/(BOLE_HT*12-54)
          DMID=(DBOT+8)/2
     ELSE
          DBOT=0
          DMID=0
     ENDIF
     BT2=.48+(.042*DMID)
     VF=(-2.4068+(.1801*DBH^2))
     RUB=(DMID-BT2)/2
     VS=3.14159*(RUB/12)^2*BOLE_HT
     IF LAND='1'
          TS1=TS1+1
```

```
          TVF1=TVF1+VF
          TVS1=TVS1+VS
     ENDIF
     IF LAND='2'
          TS2=TS2+1
          TVF2=TVF2+VF
          TVS2=TVS2+VS
     ENDIF
     IF LAND='3'
          TS3=TS3+1
          TVF3=TVF3+VF
          TVS3=TVS3+VS
     ENDIF
     IF LAND='4'
          TS4=TS4+1
          TVF4=TVF4+VF
          TVS4=TVS4+VS
     ENDIF
     SKIP
ENDDO
TVF1=TVF1*CM
TVF2=TVF2*CM
TVF3=TVF3*CM
TVF4=TVF4*CM
TVS1=TVS1*CM
TVS2=TVS2*CM
TVS3=TVS3*CM
TVS4=TVS4*CM
PC1=(TVF1/POP1)*.85
PC2=TVF2/POP2
PC3=TVF3/POP3
PC4=TVF4/POP4
F1=TVS1/TVF1
F2=TVS2/TVF2
F3=TVS3/TVF3
F4=TVS4/TVF4
TVF1=SP*R1*PC1/1000
TVF2=SP*R2*PC2/1000
TVF3=SP*R3*PC3/1000
TVF4=SP*R4*PC4/1000
TVS1=TVF1*F1TVS2=TVF2*F2
TVS3=TVF3*F3
TVS4=TVF4*F4
FW1=TVF1-TVS1
FW2=TVF2-TVS2
FW3=TVF3-TVS3
FW4=TVF4-TVS4
@7,2 SAY 'Less than .5'
@7,19 SAY TVF1 PICTURE '999999'
@7,32 SAY PC1 PICTURE '9.99'
@7,43 SAY TVS1 PICTURE '999999'
@7,51 SAY F1 PICTURE '.99'
@7,58 SAY FW1 PICTURE '999999'
@9,2 SAY '.50 to <2.50'
@9,19 SAY TVF2 PICTURE '999999'
@9,32 SAY PC2 PICTURE '9.99'
@9,43 SAY TVS2 PICTURE '999999'
@9,51 SAY F2 PICTURE '.99'
@9,58 SAY FW2 PICTURE '999999'
@11,2 SAY '2.5 to <7.5'
@11,19 SAY TVF3 PICTURE '999999'
@11,32 SAY PC3 PICTURE '9.99'
@11,43 SAY TVS3 PICTURE '999999'
@11,51 SAY F3 PICTURE '.99'
@11,58 SAY FW3 PICTURE '999999'
@13,2 SAY '7.5 & above'
@13,19 SAY TVF4 PICTURE '999999'
@13,32 SAY PC4 PICTURE '9.99'
@13,43 SAY TVS4 PICTURE '999999'
@13,51 SAY F4 PICTURE '.99'
@13,58 SAY FW4 PICTURE '999999'
```

```
@15,2 SAY REPLICATE(CHR(196),63)
@17,2 SAY 'f - fraction of total volume'
IF P_P='Y'
   SET DEVICE TO SCREEN
   @0,0 SAY " "
ELSE
   DO H_SCR
ENDIF
RETURN
*:**************************************
*:    Program: PART101.FMT
*:    Called by: ENTRY1 (proc.in VFIEN.PRG)
*:**************************************
@2,5 SAY 'SAMPLE ' GET
                              PART101->SAMPLE
@2,50 SAY 'ID ' GET PART101->ID
@4,5 SAY 'FAMILY  '
@4,15 SAY 'TOTAL ' GET
                              PART101->FAMILY
@4,30 SAY 'MALE ' GET PART101->MALE
@4,45 SAY 'FEMALE ' GET
                              PART101->FEMALE
@4,60 SAY 'MINOR ' GET
                               PART101->MINOR
@6,5 SAY 'OCCUPATION ' GET
                              PART101->OCCUPA
@8,5 SAY 'LAND OWNERSHIP ' GET
                                PART101->LAND
@10,5 SAY 'LIVESTOCK '
@10,30 SAY 'CATTLE ' GET
                              PART101->CATTLE
@10,50 SAY 'BUFFALO ' GET
                             PART101->BUFFALO
@12,10 SAY 'GOAT ' GET PART101->GOAT
@12,30 SAY 'SHEEP ' GET
                               PART101->SHEEP
@12,50 SAY 'OTHERS ' GET
                            PART101->OTH_LIVE
@15,5 SAY 'CANES - NUMBER OF CLUMS'
@17,10 SAY 'GOLLA ' GET
                            PART101->CA_GOLLA
@17,30 SAY 'JALI ' GET
                              PART101->CA_JALI
@17,50 SAY 'MURTA ' GET
                            PART101->CA_MURTA
@20,10 SAY 'THATCH GRASS ' GET
                               PART101->GRASS
@1,1 TO 3,70
@3,1 TO 5,70
@5,1 TO 7,70
@7,1 TO 9,70
@9,1 TO 13,70
@14,1 TO 18,70
@19,1 TO 21,70
*:**************************************
*:    Program: PART102.FMT
*:    Called by: ENTRY1 (proc.in VFIEN.PRG)
*:**************************************
@1,10 SAY 'KATABASH: '
@1,25 GET PART102->B1_CL
@1,35 GET PART102->B1_MS
@1,45 GET PART102->B1_IS
@3,10 SAY 'BARIALA:'
@3,25 GET B2_CL
@3,35 GET B2_MS
@3,45 GET B2_IS
@5,10 SAY 'BARUA:'
@5,25 GET B3_CL
@5,35 GET B3_MS
@5,45 GET B3_IS
@7,10 SAY 'JAI:'
@7,25 GET B4_CL
@7,35 GET B4_MS
@7,45 GET B4_IS
@9,10 SAY 'MAKHAL:'
@9,25 GET B5_CL
@9,35 GET B5_MS
@9,45 GET B5_IS
@11,10 SAY 'ORAH:'
@11,25 GET B6_CL
@11,35 GET B6_MS
@11,45 GET B6_IS
@13,10 SAY 'KALISERI:'
@13,25 GET B7_CL
@13,35 GET B7_MS
@13,45 GET B7_IS
@15,10 SAY 'TARALA:'
@15,25 GET B8_CL
@15,35 GET B8_MS
@15,45 GET B8_IS
@17,10 SAY 'BARAK:'
@17,25 GET B9_CL
@17,35 GET B9_MS
@17,45 GET B9_IS
@19,10 SAY 'MITINGA:'
@19,25 GET B10_CL
@19,35 GET B10_MS
@19,45 GET B10_IS
@21,10 SAY 'MULI:'
@21,25 GET B11_CL
@21,35 GET B11_MS
@21,45 GET B11_IS
@23,10 SAY 'OTHERS:'
@23,25 GET B12_CL
@23,35 GET B12_MS
@23,45 GET B12_IS

*:**************************************
*:    Program: PART103.FMT
*:    Called by: ENTRY1 (proc. VFIEN.PRG)
*:**************************************
@8,10 SAY 'SPECIES CODE:  ' GET
                              PART103->CODE
@12,10 SAY 'NUMBER:          ' GET
                            PART103->NUMBER
@6,8 TO 14,32

*:**************************************
*:    Program: PART104.FMT
*:    Called by: ENTRY1 (proc.in VFIEN.PRG)
*:**************************************@5,10  SAY
'SPECIES CODE:'
@5,30 GET CODE
@8,10 SAY 'GIRTH:'
@8,30 GET GIRTH
@11,10 SAY 'BOLE HEIGHT:'
@11,30 GET BOLE_HT
@3,6 TO 13,38
*:**************************************
*:    Program: PART105.FMT
*:    Called by: ENTRY1 (proc.in VFIEN.PRG)
*:**************************************
@5,10 SAY 'NAME        5-10     10-20
                                         ABOVE 20'

@7,10 SAY 'TAL:'
@7,22 GET TAL1
@7,32 GET TAL2
@7,42 GET TAL3
@9,10 SAY 'NARKEL:'
@9,22 GET NAR1
@9,32 GET NAR2
@9,42 GET NAR3
```

```
@11,10 SAY 'KHEJUR:'
@11,22 GET KHE1
@11,32 GET KHE2
@11,42 GET KHE3
@13,10 SAY 'SUPARI:'
@13,22 GET SUP1
@13,32 GET SUP2
@13,42 GET SUP3
@3,6 TO 15,52
*:**************************************
*:    Program: PART106.FMT
*:    Called by: ENTRY1 (proc.in VFIEN.PRG)
*:**************************************
@5,10 SAY 'NAME     BELOW 5    5-10
                                   ABOVE 10'
@7,10 SAY 'PEARA:'
@7,22 GET PEA1
@7,32 GET PEA2
@7,42 GET PEA3
@9,10 SAY 'LEBU:'
@9,22 GET LEB1
@9,32 GET LEB2
@9,42 GET LEB3
@11,10 SAY 'BANANA:'
@11,22 GET BAN1
@11,32 GET BAN2
@11,42 GET BAN3
@13,10 SAY 'OTHERS:'
@13,22 GET OTH1
@13,32 GET OTH2
@13,42 GET OTH3
@3,6 TO 15,52
*:**************************************
*:    Program: PART201.FMT
*:    Called by: ENTRY2 (proc. VFIEN.PRG)
*:**************************************
@1,1 SAY 'ID ' GET ID
@1,18 SAY '1     2     3     4     5     6'
@3,1 SAY 'FIREWOOD'
@3,17 GET R1C1
@3,24 GET R1C2
@3,32 GET R1C3
@3,40 GET R1C4
@3,48 GET R1C5
@3,56 GET R1C6
@5,1 SAY 'BRANCHES'
@5,17 GET R2C1
@5,24 GET R2C2
@5,32 GET R2C3
@5,40 GET R2C4
@5,48 GET R2C5
@5,56 GET R2C6
@7,1 SAY 'TREE WASTE'
@7,17 GET R3C1
@7,24 GET R3C2
@7,32 GET R3C3
@7,40 GET R3C4
@7,48 GET R3C5
@7,56 GET R3C6
@9,1 SAY 'BAMBOO'
@9,17 GET R4C1
@9,24 GET R4C2
@9,32 GET R4C3
@9,40 GET R4C4
@9,48 GET R4C5
@9,56 GET R4C6
@11,1 SAY 'AGRI RESIDUE'
@11,17 GET R5C1
@11,24 GET R5C2
@11,32 GET R5C3
@11,40 GET R5C4
```

```
@11,48 GET R5C5
@11,56 GET R5C6
@13,1 SAY 'COW DUNG'
@13,17 GET R6C1
@13,24 GET R6C2
@13,32 GET R6C3
@13,40 GET R6C4
@13,48 GET R6C5
@13,56 GET R6C6
@15,1 SAY 'CHARCOAL'
@15,17 GET R7C1
@15,24 GET R7C2
@15,32 GET R7C3
@15,40 GET R7C4
@15,48 GET R7C5
@15,56 GET R7C6
@17,1 SAY 'OIL, GAS, ELEC'
@17,17 GET R8C1
@17,24 GET R8C2
@17,32 GET R8C3
@17,40 GET R8C4
@17,48 GET R8C5
@17,56 GET R8C6
@2,0 TO 18,58
READ
CLEAR
@2,10 SAY 'FUEL'
@4,10 SAY '(b)....' GET FU_B
@6,10 SAY '(c)....' GET FU_C
@8,10 SAY '(d)....' GET FU_D
@10,10 SAY '(c)....' GET FU_E
@10,22 GET FU_E1
@12,10 SAY '(f)....' GET FU_F
@14,10 SAY '(g)....' GET FU_H
@16,10 SAY '(h)....' GET FU_H
@18,10 SAY '(i)....' GET FU_I
@1,9 TO 19,24
@2,40 SAY 'FODDER'
@4,40 SAY '(a)....' GET FO_A
@6,40 SAY '(b)....' GET FO_B
@8,40 SAY '(c)....' GET FO_C
@10,40 SAY '(d)....' GET FO_D
@12,40 SAY '(e)....' GET FO_E
@14,40 SAY '(f)....' GET FO_F
@16,40 SAY '(g)....' GET FO_G
@16,52 GET FO_G1
@18,40 SAY '(h)....' GET FO_H
@20,40 SAY '(i)....' GET FO_I
@1,39 TO 21,53
*:**************************************
*:    Program: PART202.FMT
*:    Called by: ENTRY2 (proc.in VFIEN.PRG)
*:**************************************
@2,5 SAY 'MAIN BUILDING MATERIALS'
@4,16 SAY '1      2      3      4      5
                                           6      7'
@5,5 SAY 'ROOFS'
@5,15 GET ROO1
@5,22 GET ROO2
@5,30 GET ROO3
@5,38 GET ROO4
@5,45 GET ROO5
@5,51 GET ROO51
@5,60 GET ROO6
@5,67 GET ROO7
@7,5 SAY 'CEILINGS'
@7,15 GET CEI1
@7,22 GET CEI2
@7,30 GET CEI3
@7,38 GET CEI4
@7,45 GET CEI5
```

```
@7,51 GET CEI51
@7,60 GET CEI6
@7,67 GET CEI7
@9,5 SAY 'WALLS'
@9,15 GET WAL1
@9,22 GET WAL2
@9,30 GET WAL3
@9,38 GET WAL4
@9,45 GET WAL5
@9,51 GET WAL51
@9,60 GET WAL6
@9,67 GET WAL7
@3,4 TO 10,71
@14,16 SAY ' 1       2       3       4
                                                      5       6'
@15,5 SAY 'DOORS'
@15,15 GET DOO1
@15,23 GET DOO2
@15,31 GET DOO3
@15,38 GET DOO4
@15,44 GET DOO41
@15,52 GET DOO5
@15,59 GET DOO6
@17,5 SAY 'PILLARS'
@17,15 GET PIL1
@17,23 GET PIL2
@17,31 GET PIL3
@17,38 GET PIL4
@17,44 GET PIL41
@17,52 GET PIL5
@17,59 GET PIL6
@19,5 SAY 'WINDOWS'
@19,15 GET WIN1
@19,23 GET WIN2
@19,31 GET WIN3
@19,38 GET WIN4
@19,44 GET WIN41
@19,52 GET WIN5
@19,59 GET WIN6
@13,4 TO 20,63
READ
CLEAR
@2,5 SAY 'OTHER BUILDING
                                              MATERIALS'
@4,16 SAY ' 1       2       3       4       5'
@5,5 SAY 'KITCHENS'
@5,15 GET KIT1
@5,23 GET KIT2
@5,31 GET KIT3
@5,38 GET KIT4
@5,44 GET KIT41
@5,53 GET KIT5
@7,5 SAY 'LATRINES'
@7,15 GET LAT1
@7,23 GET LAT2
@7,31 GET LAT3
@7,38 GET LAT4
@7,44 GET LAT41
@7,53 GET LAT5
@9,5 SAY 'OTHERS'
@9,15 GET OTH1
@9,23 GET OTH2
@9,31 GET OTH3
@9,38 GET OTH4
@9,44 GET OTH41
@9,53 GET OTH5
@3,4 TO 10,57
@14,5 SAY 'FENCING'
@16,11 SAY '1       2       3       4       5       6'
@17,10 GET FEN1
@17,18 GET FEN2
```

```
@17,25 GET FEN3
@17,33 GET FEN4
@17,40 GET FEN5
@17,47 GET FEN6
@15,4 TO 18,53
*:*************************************
*:     Program: PART203.FMT
*:     Called by: ENTRY2 (proc.in VFIEN.PRG)
*:*************************************
@2,2 SAY 'BED COT - std'
@2,20 GET BED_ST1
@2,26 GET BED_ST2
@4,2 SAY 'BED COT - dec'
@4,20 GET BED_DE1
@4,26 GET BED_DE2
@6,2 SAY 'TABLE - drawer'
@6,20 GET TAB_DR1
@6,26 GET TAB_DR2
@8,2 SAY 'TABLE - normal'
@8,20 GET TAB_NO1
@8,26 GET TAB_NO2
@10,2 SAY 'CHAIR - armless'
@10,20 GET CHA_AL1
@10,26 GET CHA_AL2
@12,2 SAY 'CHAIR with arm'
@12,20 GET CHA_AR1
@12,26 GET CHA_AR2
@14,2 SAY 'CHAIR-cane seat'
@14,20 GET CHA_CS1
@14,26 GET CHA_CS2
@16,2 SAY 'ALMIRA - no gla'
@16,20 GET ALM_NG1
@16,26 GET ALM_NG2
@18,2 SAY 'ALMIRAH'
@18,20 GET ALMIRA1
@18,26 GET ALMIRA2
@20,2 SAY 'BENCH - high'
@20,20 GET BEN_HI1
@20,26 GET BEN_HI2
@1,1 TO 21,28
@2,40 SAY 'BENCH - seat'
@2,58 GET BEN_SE1
@2,64 GET BEN_SE2
@4,40 SAY 'BENCH-arm & back'
@4,58 GET BEN_AB1
@4,64 GET BEN_AB2
@6,40 SAY 'SHELF'
@6,58 GET SHELF1
@6,64 GET SHELF2
@8,40 SAY 'PIRA'
@8,58 GET PIRA1
@8,64 GET PIRA2
@10,40 SAY 'BOX'
@10,58 GET BOX1
@10,64 GET BOX2
@12,40 SAY 'ALNA'
@12,58 GET ALNA1
@12,64 GET ALNA2
@14,40 SAY 'CHOWKI'
@14,58 GET CHOWKI1
@14,64 GET CHOWKI2
@16,40 SAY 'CHOWKI-single'
@16,58 GET CHO_SI1
@16,64 GET CHO_SI2
@18,40 SAY 'DESK'
@18,58 GET DESK1
@18,64 GET DESK2
@20,40 SAY 'STOOL'
@20,58 GET STOOL1
@20,64 GET STOOL2
@1,39 TO 21,66
```

```
READ
CLEAR
@1,1 SAY 'CANE FURNITURE'
@3,10 SAY 'LAWN CHAIR-single'
@3,30 GET LCH_SI1
@3,38 GET LCH_SI2
@3,46 GET LCH_SI3
@5,10 SAY 'LAWN CHAIR-double'
@5,30 GET LCH_DO1
@5,38 GET LCH_DO2
@5,46 GET LCH_DO3
@7,10 SAY 'TABLE - central'
@7,30 GET TAB_CE1
@7,38 GET TAB_CE2
@7,46 GET TAB_CE3
@9,10 SAY 'TABLE - side'
@9,30 GET TAB_SI1
@9,38 GET TAB_SI2
@9,46 GET TAB_SI3
@11,10 SAY 'MURA - round'
@11,30 GET MUR_RO1
@11,38 GET MUR_RO2
@11,46 GET MUR_RO3
@13,10 SAY 'MURA - chair'
@13,30 GET MUR_CH1
@13,38 GET MUR_CH2
@13,46 GET MUR_CH3
@15,10 SAY 'SUITCASE'
@15,30 GET SUIT1
@15,38 GET SUIT2
@15,46 GET SUIT3
@17,10 SAY 'CRADLE'
@17,30 GET CRADLE1
@17,38 GET CRADLE2
@17,46 GET CRADLE3
@2,9 TO 18,48
*:***************************************
*:    Program: PART204.FMT
*:    Called by: ENTRY2 (proc.in VFIEN.PRG)
*:***************************************
@1,1 SAY 'AGRICULTURAL'
@2,1 SAY 'IMPLEMENTS'
@2,18 SAY 'PLOUGH'
@2,32 GET PLOUGH1
@2,41 GET PLOUGH2
@2,47 GET PLOUGH3
@4,18 SAY 'LADDER'
@4,32 GET LADDER1
@4,41 GET LADDER2
@4,47 GET LADDER3
@6,18 SAY 'DHEKI'
@6,32 GET DHEKI1
@6,41 GET DHEKI2
@6,47 GET DHEKI3
@8,18 SAY 'RICE POUNDER'
@8,32 GET RICE1
@8,41 GET RICE2
@8,47 GET RICE3
@10,18 SAY 'SPADE'
@10,32 GET SPADE1
@10,41 GET SPADE2
@10,47 GET SPADE3
@12,18 SAY 'POLO'
@12,32 GET POLO1
@12,41 GET POLO2
@12,47 GET POLO3
@14,18 SAY 'TOPA'
@14,32 GET TOPA1
@14,41 GET TOPA2
@14,47 GET TOPA3
@16,18 SAY 'JHAKA'
@16,32 GET JHAKA1
@16,41 GET JHAKA2
@16,47 GET JHAKA3
@18,18 SAY 'KHOLUI'
@18,32 GET KHOLUI1
@18,41 GET KHOLUI2
@18,47 GET KHOLUI3
@20,18 SAY 'GRAIN BIN'
@20,32 GET GRAIN1
@20,41 GET GRAIN2
@20,47 GET GRAIN3
@1,17 TO 21,50
*:***************************************
*:    Program: PART205.FMT
*:    Called by: P2ENTRY (proc. VFIEN.PRG)
*:***************************************
@1,1 SAY 'TRANSPORTATION'
@2,25 SAY ' 1      2      3      4'
@4,10 SAY 'BOAT-BELOW 15'
@4,26 GET BOAT_S1
@4,33 GET BOAT_S2
@4,42 GET BOAT_S3
@4,49 GET BOAT_S4
@6,10 SAY 'BOAT-15 TO 30'
@6,26 GET BOAT_M1
@6,33 GET BOAT_M2
@6,42 GET BOAT_M3
@6,49 GET BOAT_M4
@8,10 SAY 'BOAT-ABOVE 30'
@8,26 GET BOAT_L1
@8,33 GET BOAT_L2
@8,42 GET BOAT_L3
@8,49 GET BOAT_L4
@10,10 SAY 'CART'
@10,26 GET CART1
@10,33 GET CART2
@10,42 GET CART3
@10,49 GET CART4
@12,10 SAY 'RICKSHAW'
@12,26 GET RICK1
@12,33 GET RICK2
@12,42 GET RICK3
@12,49 GET RICK4
@14,10 SAY 'DULEE'
@14,26 GET DULEE1
@14,33 GET DULEE2
@14,42 GET DULEE3
@14,49 GET DULEE4
@16,10 SAY 'PALKI'
@16,26 GET PALKI1
@16,33 GET PALKI2
@16,42 GET PALKI3
@16,49 GET PALKI4
@18,10 SAY 'OTHERS'
@18,26 GET OTHER1
@18,33 GET OTHER2
@18,42 GET OTHER3
@18,49 GET OTHER4
@3,9 TO 19,51
*:***************************************
*:    Program: PART206.FMT
*:    Called by:P2ENTRY(proc.in VFIEN.PRG)
*:***************************************
@1,1 SAY 'SALES & PURCHASE'
@3,20 SAY ' 1      2     3      4      5     6'
@5,2 SAY 'TIMBER round'
@5,20 GET TIM_R1
@5,29 GET TIM_R2
@5,36 GET TIM_R3
@5,41 GET TIM_R4
@5,50 GET TIM_R5
```

```
@5,57 GET TIM_R6
@7,2 SAY 'TIMBER sawn'
@7,20 GET TIM_S1
@7,29 GET TIM_S2
@7,36 GET TIM_S3
@7,41 GET TIM_S4
@7,50 GET TIM_S5
@7,57 GET TIM_S6
@9,2 SAY 'FUELWOOD'
@9,20 GET FUEL1
@9,29 GET FUEL2
@9,36 GET FUEL3
@9,41 GET FUEL4
@9,50 GET FUEL5
@9,57 GET FUEL6
@11,2 SAY 'BAMBOO'
@11,20 GET BAMB1
@11,29 GET BAMB2
@11,36 GET BAMB3
@11,41 GET BAMB4
@11,50 GET BAMB5
@11,57 GET BAMB6
@4,1 TO 12,59
*:************************************
*:   Program: PART207.FMT
*:   Called by: P2ENTRY(proc. VFIEN.PRG)
*:************************************
@5,10 SAY 'HARVEST'
@8,20 SAY 'SPECIES      GIRTH
                                        NUMBER'
@10,23 GET CODE
@10,34 GET GIRTH
@10,45 GET NUMBER
@7,19 TO 11,49

*:************************************
*:   Program: PART301.FMT
*: · Called by: ENTRY2(proc.in VFIEN.PRG)
*:************************************
@1,1 SAY 'ATTITUDE'
@3,2 SAY '1. (a)....' GET N1A
@5,2 SAY '   (b)....' GET N1B
@7,2 SAY '   (c)....' GET N1C
@2,4 TO 8,15
@3,29 SAY '2.'
@3,32 SAY 'FRUIT '
@3,42 GET N2_R1C1
@3,48 GET N2_R1C2
@3,54 GET N2_R1C3
@5,32 SAY 'TIMBER'
@5,42 GET N2_R2C1
@5,48 GET N2_R2C2
@5,54 GET N2_R2C3
@7,32 SAY 'BAMBOO'
@7,42 GET N2_R3C1
@7,48 GET N2_R3C2
@7,54 GET N2_R3C3
@2,31 TO 8,59
@12,2 SAY '3. ' GET N3
@14,6 SAY 'This year'
@14,28 GET N3_R1C1
@14,35 GET N3_R1C2
@16,6 SAY 'Last year'
@16,28 GET N3_R2C1
@16,35 GET N3_R2C2
@18,6 SAY 'Total last 3 yrs'
@18,28 GET N3_R3C1
```

```
@18,35 GET N3_R3C2
@20,6 SAY 'Total more than 3 yrs'
@20,28 GET N3_R4C1
@20,35 GET N3_R4C2
@11,4 TO 21,40
@12,50 SAY '4. (a)....' GET N4A
@14,53 SAY '(b)....' GET N4B
@16,53 SAY '(c)....' GET N4C
@18,53 SAY '(d)....' GET N4D
@11,52 TO 19,63
READ
CLEAR
@2,10 SAY '5. (a)....' GET N5A
@4,13 SAY '(b)....' GET N5B
@6,13 SAY '(c)....' GET N5C
@8,13 SAY '(d)....' GET N5D
@10,10 SAY '6. (a)....' GET N6A
@12,10 SAY '7  .......' GET N7
@14,10 SAY '8  .......' GET N8
@16,10 SAY '9. (a)....' GET N9A
@18,13 SAY '(b)....' GET N9B
@1,11 TO 19,23
@2,34 SAY '10  .......' GET N10
@4,34 SAY '11  .......' GET N11
@6,34 SAY '12  .......' GET N12
@8,34 SAY '13  .......' GET N13
@10,34 SAY '14  .......' GET N14
@12,34 SAY '15  .......' GET N15
@14,34 SAY '16  .......' GET N16
@1,36 TO 15,48
@2,59 SAY '17. (a)....' GET N17A
@4,63 SAY '(b)....' GET N17B
@6,63 SAY '(c)....' GET N17C
@8,59 SAY '18. (a)....' GET N18A
@10,63 SAY '(b)....' GET N18B
@12,63 SAY '(c)....' GET N18C
@14,59 SAY '19. (a)....' GET N19A
@16,63 SAY '(b)....' GET N19B
@1,61 TO 17,73
READ
CLEAR
@2,9 SAY '20. (a)....' GET N20A
@4,13 SAY '(b)....' GET N20B
@6,13 SAY '(c)....' GET N20C
@8,13 SAY '(d)....' GET N20D
@10,9 SAY '21. (a)....' GET N21A
@12,13 SAY '(b)....' GET N21B
@14,13 SAY '(c)....' GET N21C
@16,9 SAY '22. (a)....' GET N22A
@18,13 SAY '(b)....' GET N22B
@1,11 TO 19,23
@2,34 SAY '23  .......' GET N23
@4,34 SAY '24. (a)....' GET N24A
@6,38 SAY '(b)....' GET N24B
@8,34 SAY '25. (a)....' GET N25A
@10,38 SAY '(b)....' GET N25B
@12,38 SAY '(c)....' GET N25C
@1,36 TO 13,48
@2,59 SAY '26. (a)....' GET N26A
@4,63 SAY '(b)....' GET N26B
@6,63 SAY '(c)....' GET N26C
@8,63 SAY '(d)....' GET N26D
@10,59 SAY '27  .......' GET N27
@12,59 SAY '28  .......' GET N28
@1,61 TO 13,73
************************************
```

3. DATA FILE STRUCTURE

Database: PART101.DBF
Number of data records: 6670
Date of last update: 11/04/92

Fld	Field Name	Type	Width	Dec
1	SAMPLE	Character	8	
2	ID	Character	4	
3	FAMILY	Numeric	2	
4	MALE	Numeric	2	
5	FEMALE	Numeric	2	
6	MINOR	Numeric	2	
7	OCCUPA	Character	1	
8	LAND	Character	1	
9	CATTLE	Numeric	2	
10	BUFFALO	Numeric	2	
11	GOAT	Numeric	2	
12	SHEEP	Numeric	2	
13	OTH_LIVE	Numeric	3	
14	CA_GOLLA	Numeric	2	
15	CA_JALI	Numeric	2	
16	CA_MURTA	Numeric	2	
17	GRASS	Numeric	5	
** Total **			45	

Database: PART102.DBF
Number of data records: 4835
Date of last update: 04/14/92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	B1_CL	Numeric	3	
3	B1_MS	Numeric	4	
4	B1_IS	Numeric	4	
5	B2_CL	Numeric	3	
6	B2_MS	Numeric	4	
7	B2_IS	Numeric	4	
8	B3_CL	Numeric	3	
9	B3_MS	Numeric	4	
10	B3_IS	Numeric	4	
11	B4_CL	Numeric	3	
12	B4_MS	Numeric	4	
13	B4_IS	Numeric	4	
14	B5_CL	Numeric	3	
15	B5_MS	Numeric	4	
16	B5_IS	Numeric	4	
17	B6_CL	Numeric	3	
18	B6_MS	Numeric	4	
19	B6_IS	Numeric	4	
20	B7_CL	Numeric	3	
21	B7_MS	Numeric	4	
22	B7_IS	Numeric	4	
23	B8_CL	Numeric	3	
24	B8_MS	Numeric	4	
25	B8_IS	Numeric	4	
26	B9_CL	Numeric	3	
27	B9_MS	Numeric	4	
28	B9_IS	Numeric	4	
29	B10_CL	Numeric	3	
30	B10_MS	Numeric	4	
31	B10_IS	Numeric	4	
32	B11_CL	Numeric	3	
33	B11_MS	Numeric	4	
34	B11_IS	Numeric	4	
35	B12_CL	Numeric	3	
36	B12_MS	Numeric	4	
37	B12_IS	Numeric	4	
** Total **			137	

Database: PART103.DBF
Number of data records: 36182
Date of last update: 04/14/92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	CODE	Character	2	
3	NUMBER	Numeric	3	
** Total **			10	

Database: PART104.DBF
Number of data records: 131874
Date of last update: 11/04/92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	CODE	Character	2	
3	GIRTH	Numeric	3	
4	BOLE_HT	Numeric	2	
** Total **			12	

Database: PART105.DBF
Number of data records: 5642
Date of last update: 04/14/92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	TAL1	Numeric	3	
3	TAL2	Numeric	3	
4	TAL3	Numeric	3	
5	NAR1	Numeric	3	
6	NAR2	Numeric	3	
7	NAR3	Numeric	3	
8	KHE1	Numeric	3	
9	KHE2	Numeric	3	
10	KHE3	Numeric	3	
11	SUP1	Numeric	4	
12	SUP2	Numeric	4	
13	SUP3	Numeric	4	
** Total **			44	

Database: PART106.DBF
Number of data records: 6064
Date of last update: 04/14/92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	PEA1	Numeric	3	
3	PEA2	Numeric	3	
4	PEA3	Numeric	3	
5	LEB1	Numeric	3	
6	LEB2	Numeric	3	
7	LEB3	Numeric	3	
8	BAN1	Numeric	3	
9	BAN2	Numeric	3	
10	BAN3	Numeric	3	
11	OTH1	Numeric	3	
12	OTH2	Numeric	3	
13	OTH3	Numeric	3	
** Total **			41	

Database: PART201.DBF
Number of data records: 1328
Date of last update: 06/ 14/ 92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	R1C1	Numeric	2	
3	R1C2	Numeric	3	
4	R1C3	Numeric	3	
5	R1C4	Numeric	3	
6	R1C5	Numeric	5	
7	R1C6	Character	1	
8	R2C1	Numeric	2	
9	R2C2	Numeric	3	
10	R2C3	Numeric	3	
11	R2C4	Numeric	3	
12	R2C5	Numeric	5	
13	R2C6	Character	1	
14	R3C1	Numeric	2	
15	R3C2	Numeric	3	
16	R3C3	Numeric	3	
17	R3C4	Numeric	3	
18	R3C5	Numeric	5	
19	R3C6	Character	1	
20	R4C1	Numeric	2	
21	R4C2	Numeric	3	
22	R4C3	Numeric	3	
23	R4C4	Numeric	3	
24	R4C5	Numeric	5	
25	R4C6	Character	1	
26	R5C1	Numeric	2	
27	R5C2	Numeric	3	
28	R5C3	Numeric	3	
29	R5C4	Numeric	3	
30	R5C5	Numeric	5	
31	R5C6	Character	1	
32	R6C1	Numeric	2	
33	R6C2	Numeric	3	
34	R6C3	Numeric	3	
35	R6C4	Numeric	3	
36	R6C5	Numeric	5	
37	R6C6	Character	1	
38	R7C1	Numeric	2	
39	R7C2	Numeric	3	
40	R7C3	Numeric	3	
41	R7C4	Numeric	3	
42	R7C5	Numeric	5	
43	R7C6	Character	1	
44	R8C1	Numeric	4	1
45	R8C2	Numeric	4	1
46	R8C3	Numeric	4	1
47	R8C4	Numeric	4	1
48	R8C5	Numeric	5	
49	R8C6	Character	1	
50	FU_B	Character	1	
51	FU_C	Character	1	
52	FU_D	Character	1	
53	FU_E	Numeric	2	
54	FU_E1	Numeric	1	
55	FU_F	Character	1	
56	FU_G	Character	1	
57	FU_H	Character	1	
58	FU_I	Character	4	
59	FO_A	Character	1	
60	FO_B	Character	1	
61	FO_C	Character	1	
62	FO_D	Character	1	
63	FO_E	Character	1	
64	FO_F	Character	1	
65	FO_G	Numeric	2	
66	FO_G1	Numeric	1	
67	FO_H	Character	1	
68	FO_I	Character	1	
** Total **			170	

Database: PART202.DBF
Number of data records: 1326
Date of last update: 04/ 22/ 92

Fld	Field Name	Type	Width	Dec
1	ID	Character	4	
2	ROO1	Character	3	
3	ROO2	Numeric	5	1
4	ROO3	Numeric	5	1
5	ROO4	Numeric	4	
6	ROO5	Character	5	
7	ROO51	Character	5	
8	ROO6	Character	3	
9	ROO7	Numeric	3	
10	CEI1	Character	3	
11	CEI2	Numeric	5	1
12	CEI3	Numeric	5	1
13	CEI4	Numeric	4	
14	CEI5	Character	5	
15	CEI51	Character	5	
16	CEI6	Character	3	
17	CEI7	Numeric	3	
18	WAL1	Character	3	
19	WAL2	Numeric	5	1
20	WAL3	Numeric	5	1
21	WAL4	Numeric	4	
22	WAL5	Character	5	
23	WAL51	Character	5	
24	WAL6	Character	3	
25	WAL7	Numeric	3	
26	DOO1	Numeric	5	1
27	DOO2	Numeric	5	1
28	DOO3	Numeric	3	
29	DOO4	Character	5	
30	DOO41	Character	5	
31	DOO5	Character	3	
32	DOO6	Numeric	3	
33	PIL1	Numeric	5	1
34	PIL2	Numeric	5	1
35	PIL3	Numeric	3	
36	PIL4	Character	5	
37	PIL41	Character	5	
38	PIL5	Character	3	
39	PIL6	Numeric	3	
40	WIN1	Numeric	5	1
41	WIN2	Numeric	5	1
42	WIN3	Numeric	3	
43	WIN4	Character	5	
44	WIN41	Character	5	
45	WIN5	Character	3	
46	WIN6	Numeric	3	
47	KIT1	Numeric	5	1
48	KIT2	Numeric	5	1
49	KIT3	Numeric	3	
50	KIT4	Character	5	
51	KIT41	Character	5	
52	KIT5	Numeric	3	
53	LAT1	Numeric	5	1
54	LAT2	Numeric	5	1
55	LAT3	Numeric	3	
56	LAT4	Character	5	
57	LAT41	Character	5	
58	LAT5	Numeric	3	
59	OTH1	Numeric	5	1

```
60  OTH2         Numeric      5    1
61  OTH3         Numeric      3
62  OTH4         Character    5
63  OTH41        Character    5
64  OTH5         Numeric      3
65  FEN1         Character    4
66  FEN2         Numeric      3
67  FEN3         Numeric      5    1
68  FEN4         Numeric      4    1
69  FEN5         Numeric      3
70  FEN6         Numeric      4

** Total **                 292
```

```
Database: PART203.DBF
Number of data records:  1316
Date of last update:   05/05/92

Fld  Field Name   Type      Width  Dec

  1  ID           Character    4
  2  BED_ST1      Numeric      2
  3  BED_ST2      Numeric      1
  4  BED_DE1      Numeric      2
  5  BED_DE2      Numeric      1
  6  TAB_DR1      Numeric      2
  7  TAB_DR2      Numeric      1
  8  TAB_NO1      Numeric      2
  9  TAB_NO2      Numeric      1
 10  CHA_AL1      Numeric      2
 11  CHA_AL2      Numeric      1
 12  CHA_AR1      Numeric      2
 13  CHA_AR2      Numeric      1
 14  CHA_CS1      Numeric      2
 15  CHA_CS2      Numeric      1
 16  ALM_NG1      Numeric      2
 17  ALM_NG2      Numeric      1
 18  ALMIRA1      Numeric      2
 19  ALMIRA2      Numeric      1
 20  BEN_HI1      Numeric      2
 21  BEN_HI2      Numeric      1
 22  BEN_SE1      Numeric      2
 23  BEN_SE2      Numeric      1
 24  BEN_AB1      Numeric      2
 25  BEN_AB2      Numeric      1
 26  SHELF1       Numeric      2
 27  SHELF2       Numeric      1
 28  PIRA1        Numeric      2
 29  PIRA2        Numeric      1
 30  BOX1         Numeric      2
 31  BOX2         Numeric      1
 32  ALNA1        Numeric      2
 33  ALNA2        Numeric      1
 34  CHOWKI1      Numeric      2
 35  CHOWKI2      Numeric      1
 36  CHO_SI1      Numeric      2
 37  CHO_SI2      Numeric      1
 38  DESK1        Numeric      2
 39  DESK2        Numeric      1
 40  STOOL1       Numeric      2
 41  STOOL2       Numeric      1
 42  LCH_SI1      Numeric      4    1
 43  LCH_SI2      Numeric      2
 44  LCH_SI3      Numeric      1
 45  LCH_DO1      Numeric      4    1
 46  LCH_DO2      Numeric      2
 47  LCH_DO3      Numeric      1
 48  TAB_CE1      Numeric      4    1
 49  TAB_CE2      Numeric      2
 50  TAB_CE3      Numeric      1
 51  TAB_SI1      Numeric      4    1
 52  TAB_SI2      Numeric      2
 53  TAB_SI3      Numeric      1
 54  MUR_RO1      Numeric      4    1
 55  MUR_RO2      Numeric      2
 56  MUR_RO3      Numeric      1
 57  MUR_CH1      Numeric      4    1
 58  MUR_CH2      Numeric      2
 59  MUR_CH3      Numeric      1
 60  SUIT1        Numeric      4    1
 61  SUIT2        Numeric      2
 62  SUIT3        Numeric      1
 63  CRADLE1      Numeric      4    1
 64  CRADLE2      Numeric      2
 65  CRADLE3      Numeric      1

** Total **                  121
```

```
Database: PART204.DBF
Number of data records:  1291
Date of last update:   04/22/92

Fld  Field Name   Type      Width  Dec

  1  ID           Character    4
  2  PLOUGH1      Numeric      5    1
  3  PLOUGH2      Numeric      2
  4  PLOUGH3      Numeric      2
  5  LADDER1      Numeric      5    1
  6  LADDER2      Numeric      2
  7  LADDER3      Numeric      2
  8  DHEKI1       Numeric      5    1
  9  DHEKI2       Numeric      2
 10  DHEKI3       Numeric      2
 11  RICE1        Numeric      5    1
 12  RICE2        Numeric      2
 13  RICE3        Numeric      2
 14  SPADE1       Numeric      5    1
 15  SPADE2       Numeric      2
 16  SPADE3       Numeric      2
 17  POLO1        Numeric      5    1
 18  POLO2        Numeric      2
 19  POLO3        Numeric      2
 20  TOPA1        Numeric      5    1
 21  TOPA2        Numeric      2
 22  TOPA3        Numeric      2
 23  JHAKA1       Numeric      5    1
 24  JHAKA2       Numeric      2
 25  JHAKA3       Numeric      2
 26  KHOLUI1      Numeric      5    1
 27  KHOLUI2      Numeric      2
 28  KHOLUI3      Numeric      2
 29  GRAIN1       Numeric      6    1
 30  GRAIN2       Numeric      2
 31  GRAIN3       Numeric      2

** Total **                   96
```

```
Database: PART205.DBF
Number of data records:  299
Date of last update:   05/07/92

Fld  Field Name   Type      Width  Dec

  1  ID           Character    4
  2  BOAT_S1      Numeric      5    1
  3  BOAT_S2      Numeric      5    1
  4  BOAT_S3      Numeric      2
  5  BOAT_S4      Numeric      1
  6  BOAT_M1      Numeric      5    1
  7  BOAT_M2      Numeric      5    1
  8  BOAT_M3      Numeric      2
  9  BOAT_M4        Numeric       1
 10  BOAT_L1        Numeric       5      1
 11  BOAT_L2        Numeric       5      1
 12  BOAT_L3        Numeric       2
 13  BOAT_L4        Numeric       1
 14  CART1          Numeric       5      1
 15  CART2          Numeric       5      1
 16  CART3          Numeric       2
 17  CART4          Numeric       1
 18  RICK1          Numeric       5      1
 19  RICK2          Numeric       5      1
 20  RICK3          Numeric       2
 21  RICK4          Numeric       1
 22  DULEE1         Numeric       5      1
 23  DULEE2         Numeric       5      1
 24  DULEE3         Numeric       2
 25  DULEE4         Numeric       1
 26  PALKI1         Numeric       5      1
 27  PALKI2         Numeric       5      1
 28  PALKI3         Numeric       2
 29  PALKI4         Numeric       1
 30  OTHER1         Numeric       5      1
 31  OTHER2         Numeric       5      1
 32  OTHER3         Numeric       2
 33  OTHER4         Numeric       1

** Total **                            109
```

```
Database: PART206.DBF
Number of data records:   799
Date of last update:   04/22/92

Fld  Field Name     Type        Width   Dec

  1  ID             Character       4
  2  TIM_R1         Numeric         5     1
  3  TIM_R2         Numeric         3
  4  TIM_R3         Character       1
  5  TIM_R4         Numeric         5     1
  6  TIM_R5         Numeric         3
  7  TIM_R6         Character       1
  8  TIM_S1         Numeric         5     1
  9  TIM_S2         Numeric         3
 10  TIM_S3         Character       1
 11  TIM_S4         -Numeric        5     1
 12  TIM_S5         Numeric         3
 13  TIM_S6         Character       1
 14  FUEL1          Numeric         6     1
 15  FUEL2          Numeric         3     1
 16  FUEL3          Character       1
 17  FUEL4          Numeric         6     1
 18  FUEL5          Numeric         3     1
 19  FUEL6          Character       1
 20  BAMB1          Numeric         4
 21  BAMB2          Numeric         3
 22  BAMB3          Character       1
 23  BAMB4          Numeric         5     1
 24  BAMB5          Numeric         3
 25  BAMB6          Character       1

** Total **                        78
```

```
Database: PART207.DBF
Number of data records:   888
Date of last update:   04/23/92

Fld  Field Name     Type        Width   Dec

  1  ID             Character       4
  2  CODE           Character       2
  3  GIRTH          Numeric         2
  4  NUMBER         Numeric         3

** Total **                        12
```

```
Database: PART301.DBF
Number of data records:   1328
Date of last update:   04/22/92

Fld  Field Name     Type        Width   Dec

  1  ID             Character       4
  2  N1A            Character       1
  3  N1B            Character       1
  4  N1C            Character       1
  5  N2_R1C1        Numeric         4
  6  N2_R1C2        Numeric         4
  7  N2_R1C3        Numeric         4
  8  N2_R2C1        Numeric         4
  9  N2_R2C2        Numeric         4
 10  N2_R2C3        Numeric         4
 11  N2_R3C1        Numeric         4
 12  N2_R3C2        Numeric         4
 13  N2_R3C3        Numeric         4
 14  N3             Character       1
 15  N3_R1C1        Numeric         4
 16  N3_R1C2        Numeric         4
 17  N3_R2C1        Numeric         4
 18  N3_R2C2        Numeric         4
 19  N3_R3C1        Numeric         4
 20  N3_R3C2        Numeric         4
 21  N3_R4C1        Numeric         4
 22  N3_R4C2        Numeric         4
 23  N4A            Character       1
 24  N4B            Character       1
 25  N4C            Character       1
 26  N4D            Character       1
 27  N5A            Character       1
 28  N5B            Character       1
 29  N5C            Character       1
 30  N5D            Character       1
 31  N6A            Character       1
 32  N7             Character       1
 33  N8             Character       1
 34  N9A            Character       1
 35  N9B            Character       1
 36  N10            Character       1
 37  N11            Character       1
 38  N12            Character       1
 39  N13            Character       1
 40  N14            Character       1
 41  N15            Character       1
 42  N16            Character       1
 43  N17A           Character       1
 44  N17B           Character       1
 45  N17C           Character       1
 46  N18A           Character       1
 47  N18B           Character       1
 48  N18C           Character       1
 49  N19A           Character       1
 50  N19B           Character       1
 51  N20A           Character       1
 52  N20B           Character       1
 53  N20C           Character       1
 54  N20D           Character       1
 55  N21A           Character       1
 56  N21B           Character       1
 57  N21C           Character       1
 58  N22A           Character       1
 59  N22B           Character       1
 60  N23            Character       1
 61  N24A
```

```
62  N24B           Character      1
63  N25A           Character      1
64  N25B           Character      1
65  N25C           Character      1
66  N26A           Character      1
67  N26B           Character      1
68  N26C           Character      1
69  N26D           Character      1
70  N27            Character      1
71  N28            Character      1

** Total **                     126

Database: PDATA.dbf
Number of data records: 7
Date of last update: 10/02/92

Fld  Field Name    Type       Width  Dec

  1  STRTA         Character      1
  2  T_POP         Numeric       10
  3  SMPL1         Numeric        6
  4  SMPL2         Numeric        6
  5  SMPL3         Numeric        6
  6  SMPL4         Numeric        6
  7  SMPLT         Numeric        6
  8  S_P1          Numeric        8
  9  S_P2          Numeric        8
 10  S_P3          Numeric        8
 11  S_P4          Numeric        8
 12  S_PT          Numeric        8
 13  D_P1          Numeric        8
 14  D_P2          Numeric        8
 15  D_P3          Numeric        8
 16  D_P4          Numeric        8
 17  D_PT          Numeric        8
 18  P_R1          Numeric        4    2
 19  P_R2          Numeric        4    2
 20  P_R3          Numeric        4    2
 21  P_R4          Numeric        4    2

** Total **                     138
```

APPENDIX 5
REFERENCES

PROJECT 372001/25
FORESTRY MASTER PLAN
BANGLADESH TA 1355-BAN

ASIAN DEVELOPMENT BANK
MANILA PHILIPPINES
DATE: 05 NOVEMBER 1992

SYSTEMS ANALYSIS

APPENDIX 5
REFERENCES

Andel,S., 1986. Forest Inventory for Management Planning, Dhaka.

Balmforth,E.G., 1988, Forest Management Plan Manual, FAO/UNDP/BGD/85/085, Dhaka.

BCAS, 1989. Forest Resources Management in Bangladesh: Issues Problems and Prospects, Dhaka.

Dochnal,E.F.. 1988, Implementation of the Forest Resource Management System (RMS), BGD/85/085. Dhaka.

Dochnal,E.F., 1986, Establishment of a Computerized Forest Management Planning System, BGD/79/017. Dhaka.

Edelman,D.J.. Byron,R.N., Manson,D.M., 1983. User's Guide to the Bangladesh Forestry Sector Simulation, UNDP/FAO Project BGD/78/010, Dhaka.

Forestry Master Plan, 1992, Forest Management, Dhaka.

Forestry Master Plan, 1992, Statistical Report, Forest Products Demand Projection, Dhaka.

Forestry Master Plan, 1992, Wood Energy, Dhaka.

Forestry Master Plan, 1992, Statistical Report, Village Forest Inventory, Dhaka.

Hughell,D.A.. 1986, Assistance to the Forestry Sector of Bangladesh, Final Report, BGD/79/017, Dhaka.

Munaweera.D.P., 1986, Assistance to the Forestry Sector of Bangladesh, Interim Report, BGD/79/017, Dhaka.

Munaweera.D.P., 1986, Establishment of a Stand-wise Data Processing Computer Software Package for Forest Management, BGD/79/017, Dhaka.

Pelz,Dr.D.R.. Pushparajah,M., 1987, Mission Report: Review of the Forest Resources Management System, BGD 85/085, Dhaka.