(1)

United States Army
Recruiting Command

USAREC SR 90-8

ARMY RECRUITING RESOURCING SYSTEM

BY

JEFF BARNES

September 1990

Research & Studies Division
U.S. Army Recruiting Command
Program Analysis and Evaluation Directorate
Fort Sheridan, Illinois 60037-6090

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	SEPTEMBER 1990	FINAL

4. TITLE AND SUBTITLE
ARMY RECRUITING RESOURCING SYSTEM

5. FUNDING NUMBERS
DAKF15-87-D-0144
DO 0009

6. AUTHOR(S)
JEFF BARNES

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
HumRRO International, Inc. (HII)
1100 S. Washington Street
Alexandria, VA 22314

8. PERFORMING ORGANIZATION REPORT NUMBER
FR90-33

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ U.S. Army Recruiting Command
USARCPAE-RS
Ft. Sheridan, IL 60037

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
SR90-8

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release, Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The U.S. Army Recruiting Command's (USAREC) ability to recruit new soldiers in the quantity and with the quality required to sustain the Regular Army and the Army Reserve depends upon its ability to adjust to changes in the marketplace. Numerous studies have established that the supply of recruits is inversely related to employment opportunities in the civil sector, and, in varying degrees, directly related to the amount of resources employed in the recruiting function. Matching resources against market difficulty is a continuing management function. This report describes the development and operation of a prototype recruiting resourcing system. The system consists of two parts:

- o A Recruiting Difficulty Index (REDIN) times-series model to assist USAREC in forecasting changes in its recruiting market; and,
- o A spreadsheet-based Recruiting Resourcing System (RRS) that integrates the forecasted market conditions from REDIN with the resource requirement needed to achieve USAREC's mission.

The REDIN Forecasting Model is the first econometric model that obtained plausible and significant coefficients for all major recruiting resource elements. With the exception of enlistment bonuses, all variables were directly linked to budget elements in RRS.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Resource Management Econometrics Recruiting	97
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

ARMY RECRUITING RESOURCING SYSTEM

by

Jeff Barnes

September, 1990

USAREC Study Report 90-8

Research and Studies Division
Program Analysis and Evaluation Directorate
U.S. Army ecruiting Command
Fort Sheridan, IL 60037

DISCLAIMER

The views, opinions, and/or findings contained in this report are those of the author and should not be construed as an official Department of the Army policy or position, unless so designated by other documentation.

TABLE OF CONTENTS

Page

LIST OF TABLES iii

LIST OF FIGURES iii

I. INTRODUCTION i

A. Background 1

B. Recruiting Resourcing System (RRS) Study Objectives 2

C. Documenting the Resourcing Process 2

D. Development of the Recruiting Difficulty Index (REDIN) Forecasting Model 3

E. RRS Concept of Operation 5

II. REDIN USER INTERFACE 6

A. Design of REDIN User Interface 6

B. REDIN Menus 6

III. RRS COST MODEL 15

A. Prototype Philosophy 15

B. Spreadsheet Format 15

C. RRS Components 15

1. GSMA Forecasting Module 16
2. Budget Estimating Module 17
3. Marginal Cost Module 22

D. Sample Application 24

E. RRS Extensions 26

IV. CONCLUSIONS 27

V. REFERENCES 28

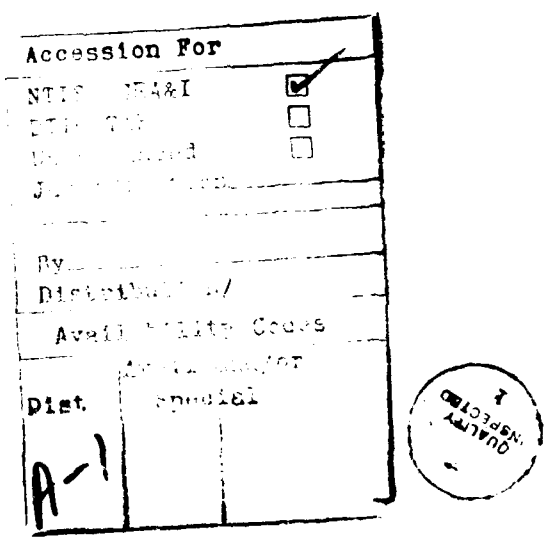
APPENDIX A - Sample RATS Output 31
APPENDIX B - REDIN Turbo Pascal Source Code 35

LIST OF TABLES

1. Variables used in GSMA model 12
2. Cost estimating factors in RRS prototype 17
3. RRS budget display 20
4. RRS marginal cost estimates 22
5. RRS summary display 23
6. RRS optimization information display 25

LIST OF FIGURES

1. Main Task Selection menu 7
2. Setup menu option 7
3. Model Selection menu 9
4. Time Series Parameters menu 10
5. Graph / Plot Options menu 11



I. INTRODUCTION

A. Background

The U.S. Army Recruiting Command's (USAREC) ability to recruit new soldiers in the quantity and with the quality required to sustain the Regular Army and the Army Reserve depends upon its ability to adjust to changes in the marketplace. Likely adjustments include (1) modifying its product offering, (2) changing the numbers of field recruiters and the locations where they are assigned, (3) promoting the Army through advertising and participation in public events, and (4) varying its enlistment terms, standards, and skill requirements. The first three factors have a direct impact on the amount of resources (both manpower and dollars expended) dedicated to the recruiting function. The fourth factor - variance of enlistment terms, standards, and skill requirements - has an indirect impact on other Army resources that are affected by changes in turnover and attrition rates, and the length of initial training required.

The reality of military recruiting is that its cost is directly related to conditions in the marketplace. Most businesses are not monopolies. Like private industry, USAREC must compete in the open market for its sales (recruits). Its competitors vary from the other military services and civilian employers to postsecondary educational institutions. But unlike private industry, USAREC presently lacks a predictor of future market conditions that has the same level of acceptance by program managers and administrators at the Department of the Army (DA), Office of the Secretary of Defense (OSD), and congressional levels as, for example, the use of the "leading," "concurrent," and "lagging" economic indicators employed by economists. Further, the federal budget process precludes the Army from using a key management program available within the private sector: a proactive marketing program wherein products, sales location, and program emphasis can be adjusted rapidly in response to changing market conditions.

Numerous studies have established that the supply of recruits is inversely related to employment opportunities in the civil sector, and, in varying degrees, directly related to the amount of resources employed in the recruiting function. These resources include the number of recruiters, advertising expenditures, type and amount of educational benefits, and enlistment bonuses. Matching resources against market difficulty is a continuing management function. The authority to allocate different mixtures of incentives and adjust resources in a fairly rapid fashion

would significantly assist recruiting management's ability to respond to changes in market conditions.

B. Recruiting Resourcing System (RRS) Study Objectives

This study had four major objectives:

- Document the system through which USAREC's recruiting mission is resourced, including the parochial interests and biases (if any) of the specific organizational elements involved in the development, review, and approval of recruiting programs and resources.

- Develop a strawman Recruiting Resourcing System that is time-sensitive and will provide USAREC enhanced ability to adjust the level of resources with changes in market conditions that were not anticipated during the program development and resourcing phases for the operating year.

- Develop a Recruiting Difficulty Index (REDIN) to assist USAREC in the forecasting of changes in its recruiting market with a level of demonstrated accuracy that should instill confidence in its ability to project future market conditions.

- Develop a computer-based Recruiting Resourcing System that integrates the forecasted market conditions from REDIN with the resource requirement needed to achieve USAREC's mission.

C. Documenting the Resourcing Process

The first two objectives were documented in an earlier report (see [29]). Based upon our interviews, we concluded that the DA staff and OSD demonstrated a solid appreciation and understanding of the importance and complexity associated with recruiting. In regard to the formal resourcing structure and management processes, HumRRO International, Inc. (HII) found no real need for changes in policies, regulations, or statutes that affect recruiting. Those interviewed within the DA staff (DCSPER, PA&E, and DAB personnel) believe there is no need for changes in the formal structure. They stated that the flexibilities within the current system and the priority given to manpower programs within the Army, and recruiting in particular, are sufficient to address any shifts in recruiting resources or product development requests. Within OSD, there is also a belief that the current system provides needed flexibility. The consensus opinion within OSD is that recruiting resource needs can be met without special accelerated

reviews or other forms of shortcuts through the resourcing process. Therefore, any proposal to overlay a special accelerated review and approval process for recruiting needs would be deemed as neither warranted nor supportable.

While this finding appears to indicate no need or desire to modify existing formal structures and processes, there were some ongoing events that pointed toward significant reductions in resource levels (e.g., Defense Management Review, unit costing for recruiting, and the potential for significant end-strength reductions). Therefore, USAREC must keep a close watch as these events develop, and be prepared to address some "worst case" scenarios. Additionally, there are a number of causes for concern in the informal structure which USAREC should address. These involve very subtle, but very powerful, informal structures and processes.

As initially conceived and described in the project management plan, RRS was envisioned as a system that combines an enlistment-forecasting model with a resource-generating model, thereby linking expected market conditions to resource requirements. To some degree, USAREC has a capability to do this by using the Rational Expectations Army Recruiting Model (REARM). However, the ability to generate resource needs based on market conditions has not been formalized with displays that are consistent with those needed for budget submissions. It is clear that the Army implementation of the Planning, Programming, Budgeting, and Execution System (PPBES) is acutely tuned to the needs and importance of recruiting. Therefore, using the terminology established in the OSD-sponsored Enlistment Early Warning System (EEWS) project,[1] the likelihood of a "recognition lag" leading to an "accession crisis" is greatly diminished. However, this does not eliminate the need for an accurate accession forecasting engine underlying the RRS. What it suggested was a little less effort toward "new" enlistment supply models (i.e., the accession forecasting engine), and more effort toward the effective linkage of resource needs to recruiting conditions and requirements. Specifically, the forecasting model must capture the impacts of the recruiting programs with significant budget allocations.

D. Development of the Recruiting Difficulty Index (REDIN) Forecasting Model

REDIN was developed under subcontract by Dr. Lawrence Goldberg of the Economic Research Laboratory (ERL). The REDIN model is fully documented in an earlier report entitled

1. Enlistment Early Warning System and Accession Crisis Prevention Process, Economic Research Laboratory Inc., Reston, VA., June 1984.

Army Recruiting Resources Study: Development of a Recruiting Difficulty Index (June 1990). Basically, REDIN is a state-of-the-art, monthly, time-series model to forecast high-quality enlistments (GSMA) at the national level. GSMA forecasting models were estimated previously by ERL for each service in a study to develop an Enlistment Early Warning System (EEWS).[2] "Transfer function" models were estimated with national monthly level data.[3] In forecasting tests the models predicted very accurately, e.g., 2% error annually, as long as new policies/programs were not introduced in the forecast period.[4] The EEWS researchers found that to achieve accurate GSMA forecasts, it was necessary to update the models soon after a new policy or program was introduced. Since this appears to happen perhaps once a year, it is necessary to update the systems regularly. For maximum accuracy, the models should be updated every month.

To forecast enlistments, it is necessary to forecast the economic factors of unemployment and civilian earnings. The EEWS also includes time-series forecasting models for these factors. Specifically, there is a transfer function model for unemployment that includes as explanatory variables 15 leading economic indicators. It also includes a model relating the growth rate of civilian earnings of youth to unemployment and the growth rate of inflation.

A limitation of the Army GSMA forecasting model included in the earlier EEWS is that it contained relatively crude measures of Army College Fund (ACF) benefits, enlistment bonuses, and advertising. The ACF benefits and bonus variables did not account for changes in the Military Occupational Specialties (MOSs) included in the programs; advertising is measured with annual rather than monthly level data. Use of poor measures of variables can result in misleading estimates of elasticities and inaccurate forecasts.

---

2. For the original research to develop the EEWS see [20]. For the most recent update of the EEWS see [16].

3. A transfer function regression model includes both explanatory variables and autoregressive or moving average error terms. For discussion, see [26].

4. If new policy/programs are introduced, it is necessary to respecify and reestimate the model to achieve accurate forecasts. This requires about four months of data in the new environment.

Building on the EEWS research, the REDIN model improved upon the earlier methodology by using improved measures of ACF benefits, bonuses, and advertising. It also includes a variable measuring the bonus "Buyup" program introduced in June 1987. The REDIN forecasting model was estimated with monthly national level data for the period October 1981-September 1989. The forecasting models for unemployment and civilian earnings of youth were also updated.

E. RRS Concept of Operation

As initially conceived, the RRS system would directly link, in a single shell, the REDIN forecasting model with a budget estimating model. However, this direct linkage was deemed unworkable for several reasons. First, REDIN is a monthly forecasting model, while budgeting is more frequently performed on an annual basis. Being a monthly model, maintaining REDIN takes time and care to update all of the input variables. Changing any single variable for a given fiscal year requires editing the 12 monthly figures that constitute the year of interest. Second, and more importantly, REDIN was written to operate on a microcomputer using the Regression Analysis of Time Series (RATS) software. Estimation of model parameters must be done in RATS. Due to the transfer function specification of REDIN (with the moving average terms), forecasting must also be accomplished in the RATS environment. Essentially, parameter estimation and forecasting are accomplished in each run. On a standard IBM-PC, this takes about 15 minutes per run. For playing "what if" drills, this would make REDIN too unresponsive to be truly useful for budget formulation.

Rather than a direct linkage we opted instead to make REDIN and RRS separate entities. RRS would include a simplified GSMA forecasting model, thereby maintaining a link between the recruiting environment, resources, and recruiting production. The REDIN and RRS forecasting equations would include the same variables with the exception of the monthly dummies and moving average terms. Unlike REDIN, RRS forecasts would be based upon *changes* in variables from a base budget year to the forecast year. Given the same equation coefficients, we would expect either model to give nearly identical forecasts. However, they would likely never be exactly equal.

Although separate, REDIN and RRS are intended to be used in tandem. REDIN is a monitoring tool; it is intended to maintain a pulse on the recruiting environment. To be of most

use, it should be maintained monthly, or no less than quarterly. RRS, on the other hand, is a planning tool. It is intended to allow budgeteers and planners to gauge the effects of changes in resource components on GSMA production and the budget. When viable options are identified, these can then be tested through REDIN to obtain more accurate GSMA forecasts.

II. REDIN USER INTERFACE

REDIN is essentially a RATS application. RATS is a very powerful language for performing times-series analyses. Unfortunately, RATS is not particularly user-friendly, nor does it possess screen utilities to build a sophisticated application user interface. Without a user interface, REDIN would require maintenance by an analyst familiar with RATS.

A. Design of REDIN User Interface

Essentially, the RATS code that constitutes the body of the REDIN program was designed to be updated by changing only a few program parameters. Although this is quite possible through directly editing and rerunning the RATS code, it would be better to have a "front-end" program that allows the user make changes in program parameters interactively. Once all desired changes are made, the program could be executed.

Due to the inherent unfriendliness of RATS, we decided to design a user interface in Turbo Pascal. Through the user interface, an analyst can change dates and output options. When the option to execute the model is selected, the Turbo Pascal program essentially writes the necessary RATS code and then calls RATS to execute this code.

B. REDIN Menus

The program name for the REDIN user interface is **USA**. When **USA** is invoked, the user is presented with a set of opening screens. When initially installed, the program will ask where the RATS program is located. The user must type the full drive and subdirectory path even if REDIN and RATS are residing in the same directory. Once this is accomplished, the program will assume the same drive and path for future model executions.

The opening menu is shown in figure 1. Throughout the program, selection of a menu option is accomplished by moving the highlight bar with the cursor keys to the desired menu option and then pressing the **ENTER** key. The first three options in the *MAIN TASK SELECTION* menu lead to submenus. When option entries are completed, the program returns to the *MAIN TASK SELECTION* menu.

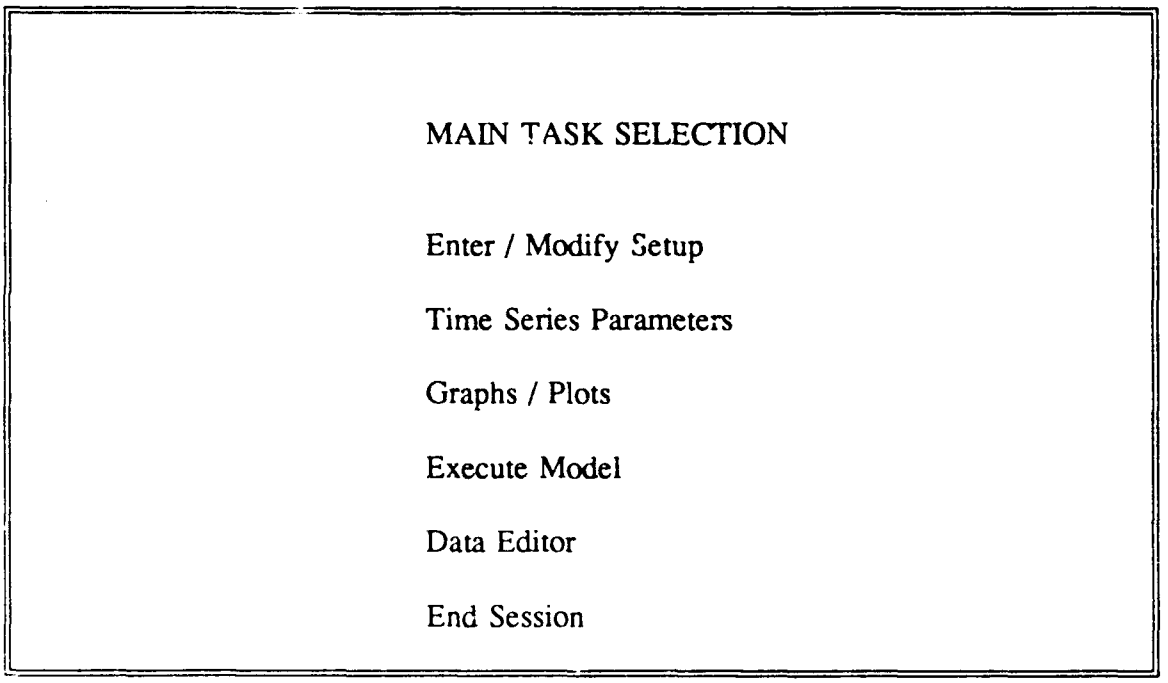


Figure 1. Main Task Selection menu

In most sessions, the first step will be to select the *ENTER / MODIFY SETUP* menu option. This submenu is shown in figure 2 below.

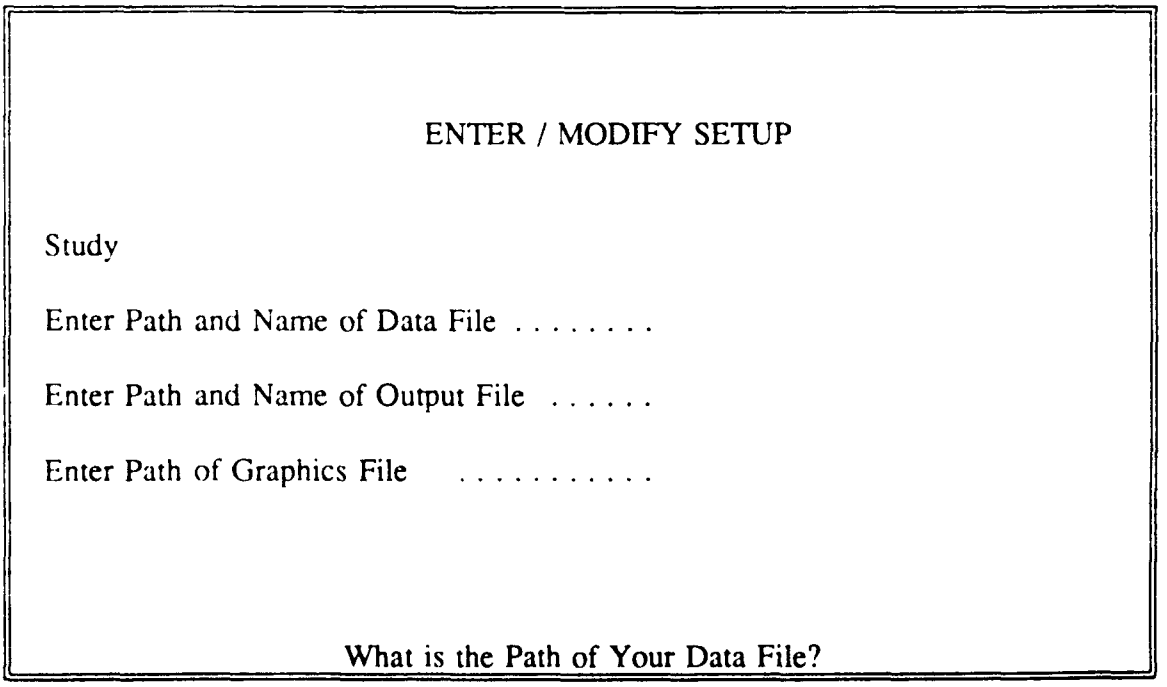


Figure 2. Setup menu option

From this menu, the user is asked for four pieces of information in sequential order. The first input requested is a study title. This is an optional entry and is only used as a label. The next input is *Enter Path and Name of Data File*. The user must provide the full drive, path, and file name. The basic input data for REDIN is contained in **DATA_RRS.RAT**. It is recommended that the user copy **DATA_RRS.RAT** to a work file (e.g., **WORK_RRS.RAT**) while performing data updates and forecasting runs. Once you are satisfied that the updated data is correct, then copy this work file back to the backup copy of **DATA_RRS.RAT**.

The next option presented is *Enter Path and Name of Output File*. The default option is to output to the screen. In most cases, output to a printer or a file will be preferred. To output to a printer, the user should enter **PRN**. To output to a file, the user should enter the full file name, including the drive and path. The last option presented is *Enter Path of Graphics File*. If desired, the graphs can be sent to a file. The alternatives are the same as the previous menu option. In most instances, the screen is the desired path.

The user will note that unique explanation prompts are displayed at the bottom of the menu as each option is reached. This is employed throughout the program to give the user a more thorough description of what information is required.

When the required information for this menu is entered, the user will be presented with the menu shown in figure 3. The REDIN system has three component models - an unemployment forecasting model, a civilian wage forecasting model, and the GSMA contracts forecasting model (referred to as REDIN). Before the user can forecast GSMA contracts, there must be forecasts of unemployment and civilian wages for the desired forecast period. In general, the update of unemployment and wages will be done once a month or quarter during routine data maintenance. As before, the user must highlight the desired option and press **ENTER**. The user will then be asked to confirm the selection. The default is to save the selected settings (Save). The user may reselect by pressing M(odify), followed by the **ENTER** key or exit the menu by pressing **Esc**. When completed, the user will be returned to the *MAIN TASK SELECTION* menu.

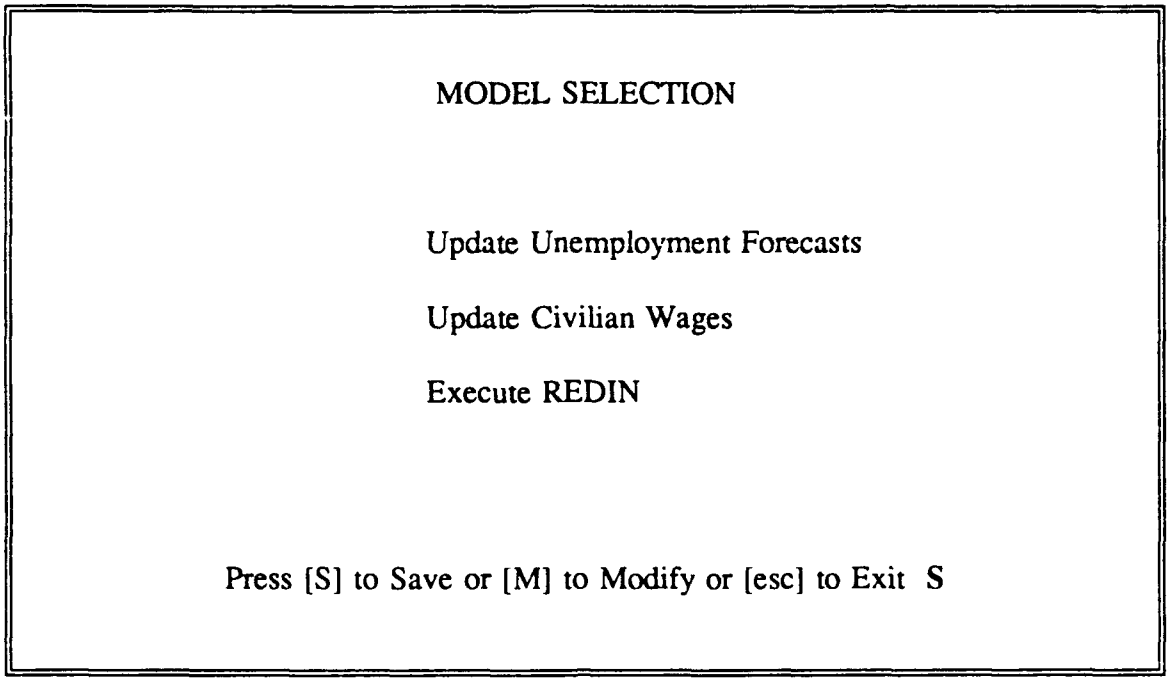


Figure 3. Model Selection menu

The second menu option in the *MAIN TASK SELECTION* menu allows the user to change the beginning and end dates for the estimation and forecast. The screen display for *Time Series Parameters* is shown in figure 4. The prototype REDIN includes data for estimating regression coefficients from FY 1981 through FY 1989 (10/80 to 9/89). The initial forecasting period is FY 1990 (10/89 to 9/90). In order to maintain the model, users must be able to extend the period for estimating the model coefficients and performing GSMA forecasts. In this prototype version, the start date for the regression must be **10/80**. Other dates may be modified as desired. However, the user must ensure that adequate data exists for the time periods specified or the program will halt with an error.

The last option of the *ENTER / MODIFY DATA SETUP FOR RATS* menu is *Graphs Only Run (Y/N)*. With this option, users can examine graphical output only. This would be a relatively infrequent need. Enter N(o) to run the full model. As in earlier menus, the user will be asked to confirm the settings before returning to the *MAIN TASK SELECTION* menu.

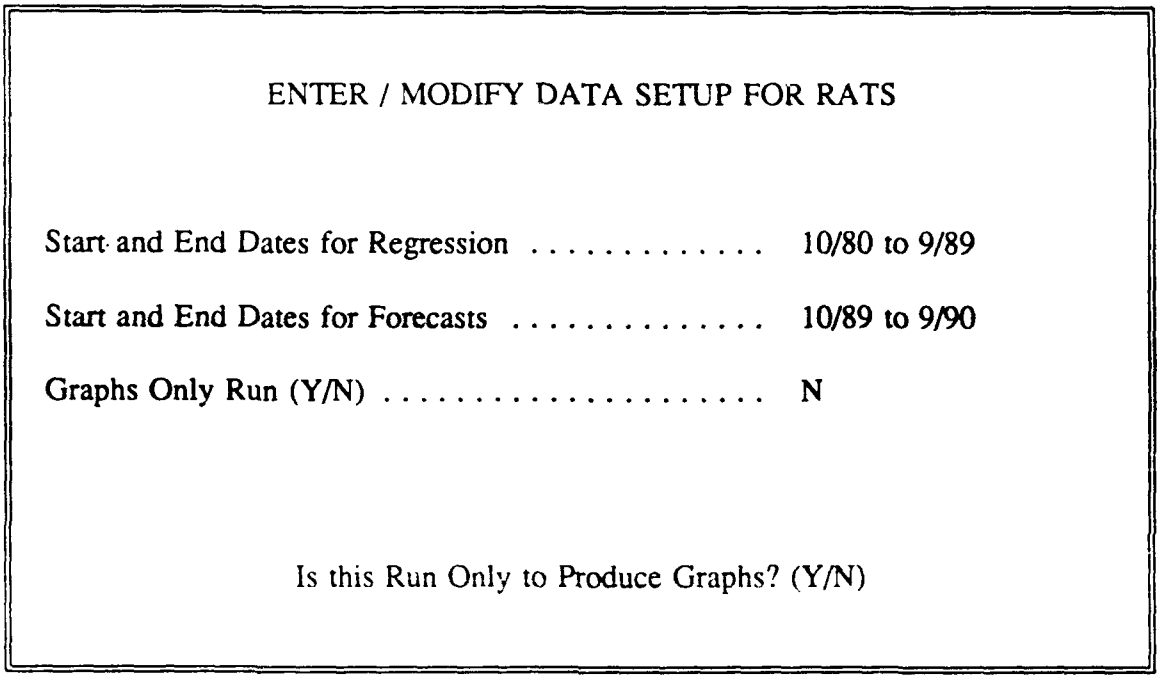


Figure 4. Time Series Parameters menu

The third option in the *MAIN TASK SELECTION* menu is *Graphs / Plots*. The *GRAPH / PLOT OPTIONS* menu is shown in figure 5. This menu allows the user to set options for graphical output. This menu is different from earlier ones in that the user must first indicate a desire to modify the default settings by pressing **M**(odify). The first option is to *Graph Data Trends*. If set to **Y**(es), the program will generate graphs of the model input data. In most instances, the graphs will reflect the input data as transformed into logarithms. Graphs of raw data can be obtained through the *Data Editor* option to be described later. The second option, *Residual Graphs*, presents a graph of the regression residuals (Calculated - Actual). This is particularly useful when updating the model to ensure that the errors for newly added periods are randomly distributed. When *Actuals vs Forecasts Graphs* is set to **Y**(es), the program will generate a graph of observed GSMA enlistments and calculated GSMA enlistments over the regression time period. In most instances, this will be the only option set to **Y**(es) in this menu. The next option is not implemented and is hard coded to **M**(onthly). Selecting the *Statistics* option turns on the printing of additional statistics. The final option, *Style (Line, Bar, Polygonal)*,

is used to specify the graph type. L(ine) is the preferred setting. The user must confirm the settings with an S(ave) before returning to the main menu.

At this point, users should be set to execute the model. When this option is chosen, the program generates the RATS code, calls and executes the RATS program, and sends the output to the specified device. A sample printout is shown in Appendix A. The output includes the estimates of the equation coefficients and a role-up of the goal, forecast, and difficulty index for the designated forecast period (usually 12 months).

GRAPH / PLOT OPTIONS

Graph Data Trends . . . . . N

Residual Graphs . . . . . N

Actuals vs. Forecasts Graphs . . . . . Y

Present Monthly, Quarterly, or Annually (M, Q, A) . . . . . M

Statistics . . . . . N

Style (Line, Bar, Polygonal) (L, B, P) L

Press [S] to Save or [M] to Modify or [esc] to Exit **S**

Figure 5. Graph / Plot Options menu

The primary task to maintain REDIN is updating the data. A list of these variables is shown in table 1. Several variables (i.e., MILPAY, BON3, ACF) require extensive preprocessing before the values can be entered into the RATS data file. Most can be entered directly. REDIN makes use of a RATS utility called **RATSDATA**. This program is fully documented in the RATS Version 3.0 manual. **RATSDATA** is a menu- and dialog-driven utility that allows users to merge several data files, extract data to create new files, list and edit contents of files, do

quick graphs of data, and convert data to various formats. When the *Data Editor* option is selected from the *MAIN TASK SELECTION* menu, REDIN invokes **RATSDATA**.

Table 1. Variables used in the GSMA models (continued on next page)

GSMA	=	(logarithm of) gross contracts for GSMA enlistments (Source: USAREC).
MILPAY	=	(logarithm of) present value (@ 30% discounted factor) of Basic Military Compensation (BMC) during a 4-year enlistment divided by the present value (@ 30% discounted factor) of earnings for 18- to -21-year-old civilian males (CIVEARN); each series is a smoothed five-month average, centered on the current month. (+) BMC, previously called regular Military Compensation, includes base pay as well as the allowances for housing and food and the tax advantage of the allowances. The calculation of BMC assumes the average time-in-grade for Army enlistees, and that the enlistee is unmarried during his/her term of service (Source: DOD/Compensation). CIVEARN is constructed with average quarterly data on the median earnings of 16- to -19 and 20- to- 24-year-old male, full-time workers (Source: Bureau of Labor Statistics from Current Population Survey).
UNEMP	=	(logarithm of) civilian unemployment rate (Source: Bureau of Labor Statistics from Current Population Survey). (+)
RECR	=	(logarithm of) Army production recruiters assigned zero, half, or full missions (Source: USAREC). (+)
GOALA	=	(logarithm of) net missions for GSMA enlistments (Source: USAREC). (+)
GOALNOTA	=	(logarithm of) net missions for non-GSMA enlistments (Source: USAREC). (-)
ADV	=	(logarithm of) sum of monthly placement expenditures for national print, radio, and TV advertising, each adjusted for inflation using separate media cost price indexes (Sources: USAREC and McKann Erickson). (+)
BONUS	=	(logarithm of) weighted average by term of service (TOS) of average bonus per taker times bonus program coverage, divided by the present value of civilian earnings of youth (CIVEARN) (Sources: Army DCSPERS, USAREC, CPS). (+)

Table 1. (Concluded)

BON3	=	for only TOS = 3, bonus per taker times bonus program coverage, divided by the present value of civilian earnings of youth (CIVEARN) (Sources: Army DCSPERS, USAREC, CPS). (+)
BON3SQ	=	BON3 squared. (-)[a]
ACF	=	(logarithm of) weighted average summed across TOSs of the present value of the maximum educational benefits available to Army enlistees (ACFPV), times ACF program coverage (ACFCOV) divided by a cost-of-college price index -- sum of tuition (four-year state university) plus room and board (Sources: Army DCSPERS, USAREC, Department of Education). (+)
POL86	=	binary variable measuring the effect of the reinstitution of the mission box in FY 1986; equal to 1.0 in FY 1986-89 and 0.0 otherwise. (+)
BUYUP	=	binary variable measuring the effects of the bonus Buyup program; equal to 1.0 in June 1987 - September 1989 and 0.0 otherwise. (+)
DUM688	=	binary variable measuring the effects of a policy change that limited production of non-GSMAs in June 1988; equal to 1.0 in June 1988 and 0.0 otherwise. (+)
DUM489	=	binary variable measuring the effects of a policy change in April 1989 that permitted greater recruitment of III-B non-HSDGs; equal to 1.0 in April 1989 and 0.0 otherwise. (-)
SEAS	=	binary monthly seasonal variables (omitted month is April).
MA(k)	=	moving average term (lag = k).

a. Negative sign expected due to the "law" of diminishing returns.

The primary use of **RATSDATA** will be to update the REDIN source data file. The first step will be to activate the *File* menu option by pressing **<Alt>F**. Use the cursor key to highlight the *Open* option and press **Enter**. The user will be asked to enter the name of the data file to edit. Enter the drive, path, and full filename (including the .RAT suffix). Next, the user must activate the *Series* menu by pressing **<Alt>S**. Use the cursor keys to highlight the *Edit Series* option and press **Enter**. The user will be presented with a list of data elements and descriptions. Move the cursor to the data element that will be edited and press the **<SPACE>** bar to highlight the variable. The data will then be displayed. Use the cursor keys to move to the month to be updated. After changes are made, the user will be asked to confirm the changes and save the data.

Another useful alternative in the *Series* option is *Graph Series*. Invoking this option produces a graph of the basic data series highlighted. The graph can be sent to the screen or a file. Using this option is a quick and easy way to view data and ensure that no large errors were made while editing the data. To exit **RATSDATA**, activate the *File* option and select *Quit*.

The final act in any REDIN session will be to *End Session*. Selecting this option returns the user to the DOS prompt. The Turbo Pascal source code for the REDIN interface is shown in Appendix B.

III. RRS COST MODEL

A. Prototype Philosophy

At the outset of this project, it was recognized that the concept of a budget adjustment mechanism beyond that currently in existence may be very difficult to sell to programmers and budgeteers. Therefore, we chose to develop a prototype RRS that demonstrates the relationships between market conditions, recruiting resources, and active enlisted recruiting success. This prototype would constitute a testbed for the concepts and form the basis for an expanded operational system. Because of its flexibility and wide base of users (particularly budget planners), the prototype system was developed using Lotus 1-2-3 (Rel. 2.01). Our primary goal was to ensure that the relationships built into the model are both accurate and useful for resource planning.

B. Spreadsheet Format

RRS is a spreadsheet model. Its use does require a basic knowledge of Lotus 1-2-3 (start the program, retrieve a file, move about in the spreadsheet, recalculate values). Lotus does have the capability to be automated (known as MACROS) to the point where users only need to know how to start the program. But, automating procedures before they are tested and accepted would be a waste of time and resources. Therefore, RRS was not extensively automated. However, it was designed with presentation screens that are both functional and sophisticated enough to instill user confidence in the technical soundness of the system.

C. RRS Components

RRS contains three major parts - the GSMA Forecast Module, the Budget Estimating Module, and the Marginal Cost Module. The basic operation of the model is one-way with a feedback loop to guide future decisions. That is, the GSMA Forecasting Module takes as inputs the environment and resource-related variables to generate a GSMA forecast. Using changes in resource variables, changes to the base year budget are calculated in the Budget Estimation Module. At this point, the analyst has the GSMA mission (basic input variable), a GSMA forecast, and a budget estimate for the forecast year. A typical application of RRS would be to add or subtract recruiters, advertising, ACF, or enlistment bonuses until missions and forecasts

are equal. To guide this decision, estimates of the marginal cost of achieving one additional GSMA contract for the four major resource variables are calculated and displayed in the Marginal Cost Module. The cost minimizing behavior would be to add resources where the marginal cost is low and subtract resources where the marginal costs are high until the marginal costs are equal.

The basic structure of RRS allows an analyst to iteratively solve for the optimal (cost minimizing) active enlisted budget to meet mission. But it does not force a solution that may be unachievable in the shortrun. The operation of each component module of RRS will be described separately.

1. GSMA Forecasting Module.

This module has five basic components. First, is the basic input data section. Included here is all of the monthly data used in the REDIN forecasting model. Data prior to the Baseline Budget Year (1989) is not currently used. The next section contains the transformed data required for the forecast. For example, the raw data for the advertising variable is the placement dollars for radio, television, and print advertisements. To calculate the variable used in the model, each raw placement dollar value is deflated by an advertising cost index, added together, and the natural logarithm of the result is calculated. Because the basic input data is monthly, and budgeting is done on an annual basis, a section of the data module is dedicated to calculating annualized base and forecast year data.

The next part contains the equation coefficients (or elasticities). The variables used for forecasting GSMA contracts in RRS are identical to those used in REDIN. As currently configured, the equation coefficients are also the same with the exception of advertising. The REDIN estimate (.02) was considered out of line as compared to results from recent battalion level time-series cross-sections (TSCS) estimates and those from prior RAND studies. Therefore, an elasticity of .05 was adopted.

The forecasting methodology used in RRS is slightly different from that used in REDIN. RRS starts with a base year. The user has the option of using the base year GSMA achievement as the starting point for the forecast or an assumed value such as the REDIN forecast for the base year. Changes in the model variables between the base and forecast year are calculated. These are then multiplied by the equation coefficients and summed. This quantity is then added to the logarithm of the base year GSMA achievement and the anti-log is calculated. Unlike REDIN,

which uses a bottoms-up calculation, the prototype RRS forecasts are based on changes in input variables from one year to the next.

2. Budget Estimating Module.

The Budget Estimating Module has two parts. The first part contains per unit cost estimates or budget element elasticities. These are shown in table 2 and were obtained from the *Cost Factor Handbook (FY 88/89)* compiled by USAREC's Resource Management and Logistics Directorate. As shown, each cost element is associated with a resource factor. The primary resource factors are recruiters, GSMA contracts, and total mission.

Table 2. Cost estimating factors in RRS prototype (continued on next page)

	Cost Factors			
				Cost/
				Elasticity
	P72		Level	
	P81		Recr	260
	P87			
	A. CIV PAY		GSMA	0.15
	B. VEHICLES		Recr	3,741
	C. BACH HOUSING		Level	
	D. COI/DEP/ED TR			
	D1. COI		Level	
	D2. DEP		Level	
	D3. ED TOURS		Level	
	D4. NURSE TOURS		Level	
	E. RCTR EXP ALLOW		Recr	720
	F. APPL M/L/T		Mission	97
	G. MIL AWARDS		Recr	
	H. BRAC		Level	

Table 2. (concluded)

	J. COPIERS		Level	
	K. ANTI-TERRORISM		Level	
	L. EQUIP RENT & PURCHASE		GSMA	0.02
	M. ADP SUMMARY			
	M1. ARADS		Level	
	M2. JOIN		Level	
	M3. OTHER ADP		Level	
	N. MISSION TRVL		GSMA	0.10
	P. RETR TRNG PRD		GSMA	0.05
	R. CIV TRNG		GSMA	0.001
	T. TOUR EXHIBITS		GSMA	0.008
	U. STUDIES		Level	
	V. FAMILY LIFE		Level	
	W. TAIR		Level	
	Z. OTHER SUMMARY			
	Z1. UPGRADE FAC		Level	
	Z2. SUPPLIES		Recr	175
	Z3. PRINTING		GSMA	0.02
	Z4. SUBSCRIPTIONS		Level	
	Z5. CONTINGENCY		Level	
	Z6.& Z8. OTHER		Level	
	Z7. POSTAGE		GSMA	0.02
	Z9. PFES		Level	
	UNDISTRIBUTED			
	RCTR SPT			
	P871712 ADVERT		Direct	

Table 2. (continued)

	INTERNS			Level	
	COMMUNICATIONS			Recr	2,000
	P87 TOTAL				
	P95			Level	
	TOTAL OMA				
	MILITARY PAY			Recr	36,769
	ENLISTMENT BONUS			Direct	
	ARMY COLLEGE FUND			Direct	

Those factors labeled "level" are candidates for manual adjustment. The factors labeled "direct" are calculated directly in the Budget Calculation section. The elasticity associated with cost elements are one way only. That is, a change in the factor does not increase the supply of GSMA recruits, but a change in GSMA recruits does change the amount needed for that cost element.

The workhorse of the Budget Estimating Module is the Budget Calculation section. This display is modeled after that used by the Resource Management Directorate in their existing spreadsheet applications. In addition, we have included cost elements other than those in the Operations and Maintenance accounts (OMA). Although not directly controlled by USAREC, these cost elements are included as part of the recruiting budget in OSD displays. Much like the forecasting module, the Budget Calculation section bases requirements in the forecast year on changes from the base year. A sample output of this spreadsheet section is shown in table 3.

Table 3. RRS budget display (continued on next page)

RECRUITING RESOURCING COST MODULE						
			Thousands of Dollars			
		BASE YEAR				Forecast
		FY 89		Delta		FY 90
P72		48				48
P81		5,176		43		5,219
P87						
A. CIV PAY		32,074		143		32,217
B. VEHICLES		26,555		617		27,172
C. BACH HOUSING		652				652
D. COI/DEP/ED TR		2,577		0		2,577
D1. COI		628				628
D2. DEP		1,076				1,076
D3. ED TOURS		737				737
D4. NURSE TOURS		136				136
E. RCTR EXP ALLOW		2,020		119		2,139
F. APPL M/L/T		12,529		(2,688)		9,841
G. MIL AWARDS		640		0		640
H. BRAC		0				0
J. COPIERS		0				0
K. ANTI-TERRORISM		0				0
L. EQUIP RENT & PURCHASE		1,681		1		1,682
M. ADP SUMMARY		27,447		0		27,447
M1. ARADS		11,553				11,553
M2. JOIN		5,530				5,530
M3. OTHER ADP		10,364				10,364

Table 3. (continued)

N. MISSION TRVL		10,026		30		10,056
P. RETR TRNG PRD		1,791		3		1,794
R. CIV TRNG		282		0		282
T. TOUR EXHIBITS		1,063		0		1,063
U. STUDIES		892				892
V. FAMILY LIFE		201				201
W. TAIR		1,450				1,450
Z. OTHER SUMMARY		7,117		30		7,147
Z1. UPGRADE FAC		285				285
Z2. SUPPLIES		2,451		29		2,480
Z3. PRINTING		428		0		428
Z4. SUBSCRIPTIONS		206				206
Z5. CONTINGENCY		2				2
Z6.& Z8. OTHER		2,720				2,720
Z7. POSTAGE		1,025		1		1,026
Z9. PFES		0				0
UNDISTRIBUTED						
RCTR SPT		$128,997		($1,747)		127,250
P871712 ADVERT		63,099		2,572		65,671
INTERNS		185				185
COMMUNICATIONS		22,695		330		23,025
P87 TOTAL		$214,976		$1,156		216,132
P95		50				50
TOTAL OMA		$220,250		$1,198		$221,448
MILITARY PAY		275,500		6,061		281,561
ENLISTMENT BONUS		42,300				42,300
ARMY COLLEGE FUND		54,300		2,657		
				0		56,957

Table 3. (concluded)

TOTAL ENLISTED PROGRAM		$592,350		$10,553		$602,903

3. Marginal Cost Module.

RRS forecasts react to changes in the recruiting environment (unemployment, civilian pay, college and media inflation). They also are influenced by four major recruiting resource elements - recruiters, advertising, ACF, and enlistment bonuses. All other things equal, increasing the supply of GSMA recruits requires increasing one or more of these resource factors. As a guide for what to change, estimates of the marginal cost of increasing the supply of GSMA contracts by one are calculated and displayed in this module. A sample display is shown in table 4.

Table 4. RRS marginal cost estimates

	Marginal Cost per GSMA		
Recruiters		$12,462	
Advertising			
Print		$12,212	
Radio		$11,356	
TV		$12,069	
Enl Bonus			
ACF(Coverage)		$6,288	

RRS Summary Display. RRS includes a section that organizes data from all parts of the spreadsheet into a single display, as shown in table 5.

Table 5. :RS summary display (continued on next page)

	RECRUITING RESOURCING COST MODULE					
		Forecast Summary				
---	---	---	---	---	---	---
		Base		Forecast		Percent
		Year		Year		Change
		========		========		========
Active Enlisted Budget		$592,350		$602,903		1.78%
GSMA Enlistments		50,896		52,407		2.97%
GSMA Goal		51,554		56,983		10.53%
Other Goal		73,129		39,927		-45.40%
Resource Variables						
Recruiters		5,789		5,954		2.85%
Print Advertising		$4,142		$3,850		-7.06%
Radio Advertising		$3,684		$4,111		11.60%
TV Advertising		$21,013		$23,451		11.60%
ACF Coverage		65.0%		68.9%		6.00%
Avg Value of ACF (PV)		$4,182		$4,182		0.00%
3-Yr Bonus Coverage		21.4%		21.4%		0.00%
Avg Value 3-Yr Bonus (PV)		$3,636		$3,991		9.75%
Military Pay (PV)		$43,980		$45,718		3.95%
Environmental Variables						

Table 5. (concluded)

Unemployment		5.25		5.32		1.33%
Civilian Pay (PV)		$33,762		$34,526		2.26%
Cost of College		$4,905		$5,121		4.40%
Print Media Index		139.56%		147.37%		5.59%
Radio Media Index		130.20%		137.04%		5.25%
TV Media Index		138.53%		145.64%		5.13%
Elasticities						
Recruiters		0.347				
GSMA Goal		0.129				
Other Goal		-0.133				
Advertising		0.050				
ACF		0.142				
Relative Pay		1.160				
Unemployment		0.762				

D. Sample Application

Assume for the moment that the end of the Cold War and budget pressures have reduced the total enlisted recruiting mission to 90,000. Based on historical guidelines, 12% of the mission will be female, 63% will be GSMAs, and no more than 5% will be male non-high school graduates. This equates to a mission of 47,400 GSMAs, 27,900 GSMBs, and 14,700 others. Our first step will be to split the spreadsheet screen into two horizontal windows. The top window should contain the Optimize display (see table 6) while the bottom window should contain the Annual Data area. Next, move to the Annual Data area and input these numbers as the missions for the forecast year and recalculate the spreadsheet.

Table 6. RRS optimization information display

Estimated Marginal Cost						
Recruiters		$12,462		Total $		$602,903
Advertising				Change $		10,553
Print		$12,212				
Radio		$11,356		GSMA Fcst		52,407
TV		$12,069		GSMA GOAL		56,983
Enl Bonus		$0				
ACF(Coverage)		$6,288				
ACF(Offering)		$11,009				

The forecast exceeds GSMA missions. Therefore, a reduction in resources is indicated. The resource factors with the highest marginal costs are recruiters and advertising (print and TV). The user would first reduce the number of recruiters by some amount (say, 400 recruiters). The marginal cost of recruiters is now less than print advertising. The user will then drop print and TV advertising until the marginal costs are about the same as for recruiters. This process continues until the GSMA forecast equals the GSMA goal. If feasible, we would recommend increasing those resource elements that have low marginal costs and reducing those with higher marginal costs until the marginal costs are equal (or nearly so). For this exercise, changing the ACF coverage or offering was assumed outside the control of USAREC.

The solution to this example results in the following resource changes:

Recruiters	5,789	5,000	(13.63%)
Print Advertising	$ 4,142	$ 3,850	(7.06%)
TV Advertising	$21,013	$19,000	(9.58%)
Radio Advertising	$ 3,684	$ 4,111	(11.6%)

The total Active Enlisted Budget is reduced by just over 11%. The final step is to perform a REDIN forecast using the recommended resource changes to confirm the RRS forecast.

E. RRS Extensions

In its current state, RRS is limited to a base year and one forecast year. Program planning involves planning over a five- to seven year horizon. Clearly, our abilities to forecast the recruiting environment beyond one year is limited. However, planners are required to make estimates beyond this short period. Therefore, they should have a tool that accommodates this need. However, development of extensions must not be out in front of development of the concepts.

Other extensions include adding graphical displays and automating the goal-seeking. With only two years of data, graphics are not required. With the addition of more forecast years, graphic displays would greatly enhance the usefulness of the system. As you add more forecast years, the interactive method of optimizing resources becomes more cumbersome. Having an automatic goal-seeking procedure would be of great benefit.

IV. CONCLUSIONS

USAREC is facing very challenging resourcing problems. The pressure exerted by the national budget deficit to make wholesale reductions in military manpower programs is strong. The only real line of defense will be budget estimates that are rationally tied to the recruiting environment and recruiting missions. RRS is a significant step in that direction.

The REDIN Forecasting Model is the first econometric model that obtained plausible and significant coefficients for all major recruiting resource elements. With the exception of enlistment bonuses, all variables were directly linked to budget elements. With the REDIN front-end, USAREC has a powerful tool to monitor the recruiting environment.

RRS and REDIN now need time to be tried. Only through real-world exercise can the concepts of RRS be brought to an operational state.

V. REFERENCES

[1] Box, G. E. P. and Jenkins, G. M. Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day, 1970.

[2] Dale, C. and Gilroy, C., "The Effects of the Business Cycle on Military Enlistment Rates." PPRG Working Paper 83-1, U.S. Army Research Institute for the Behavioral and Social Sciences, 1983.

[3] Dale, C. and Gilroy, C., "Determinants of Enlistments: A Macro economic Time-Series View," Armed Forces & Society, Vol 10, No. 2, Winter 1984.

[4] Daula, T. V. and Smith, D. A, "Estimating Enlistment Models for the U.S. Army," in R. G. Ehrenberg, ed. Research in Labor Economics, Volume 7, Greenwich, CT:JAI Press, 1985.

[5] Daula, T. V. and Smith, D. A, "Estimating Enlistment Models for the U.S. Army," Office of Economic and Manpower Analysis, U.S. Military Academy, January 1985.

[6] Dertouzos, J. N., "Recruiter Incentives and Enlistment Supply," RAND Corporation, R-3065-MIL, Santa Monica, 1984.

[7] Dertouzos, J. N.; Polich, J. M., with Bamezai, A. and Chesnutt, T., "Recruiting Effects of Army Advertising," RAND Corporation, R-3577-FMP, January, 1989.

[8] Doan, T. A. and Litterman, R. B. Regression Analysis of Time Series: User's Manual, Version 3.00. Minneapolis: VAR Econometrics, 1988.

[9] Fernandez, R. L., "Enlistment Effects and Policy Implications of the Educational Assistance Test Program," RAND Corporation, R-2-935-MRAL, September 1982.

[10] Goldberg, L. "Recruiters, Advertising, and Navy Enlistments." Center for Naval Analyses, CRC 409, November 1979.

[11] Goldberg, L., "Enlistment Supply: Past, Present and Future," Center for Naval Analyses, CNS 1168, September 1982.

[12] Goldberg, L. and Greenston, P. "New Data Collection and Variable Construction." Economic Research Laboratory, Inc., August 1986.

[13] Goldberg, L.; Greenston, P.; Goldberg, B.; and Andrews, S., "Cost Comparison of the VEAP and the New GI Bill: GICALC - A Program for Calculating the Present Value

of GI Bill Benefits," Economic Research Laboratory, Inc., Reston, Virginia, August 1986.

[14] Goldberg, L.; Greenston, P.; and Andrews, S., "Time-Series Analysis of the Effects of Educational Benefit Programs and Bonuses: The Effects of Delinking Factors," Economic Research Laboratory, Inc., Reston, Virginia, November, 1986.

[15] Goldberg, L.; Goldberg, B. S.; and Goldberg, E. A., "Forecasting Army Enlistments at the Battalion Level: A Feasibility Study," USAREC SR 87-3, Ft. Sheridan, Illinois, July 1987.

[16] Goldberg, L. and Goldberg, B., "Analysis of Military Enlistments in the 1980's," Economic Research Laboratory, Inc., Reston, Virginia, June 1988.

[17] Goldberg, L. and Freeman, R., "Study To Analyze The Cost-Effectiveness Of The ACF Program: Literature Review," Economic Research Laboratory, Inc., Reston, Virginia, August 1989.

[18] Goldberg, L. and Goldberg, B. "Effects of ACF Benefits on Army Enlistments," Economic Research Laboratory, Inc., June 1990 (forthcoming).

[19] Granger, C. W. J. Forecasting and Business in Economics. New York: Academic Press, 1980.

[20] Greenston, P.; Goldberg, L.; Hermansen, S.; and Andrews, S., "Recruitment Early Warning System, Phase II: Final Report," Vols. I and II, Economic Research Laboratory, Inc., Reston, Virginia, September 1985.

[21] Horne, D. K., "An Economic Analysis of Army Enlistment Supply," Manpower and Personnel Policy Research Group, May 1984.

[22] Jehn, C. and Shugart, W. F., "Recruiters, Quotas and the Number of Enlistments," Study 1073, Center for Naval Analysis, December 1976.

[23] Kearl, C. E.; Horne, D. K. and Gilroy, C. L. "Army Recruiting In A Tight Labor Market," Army Research Institute, 1989.

[24] Maddala, G. S., Econometrics, New York: McGraw-Hill, 1977.

[25] Morey, R. C.; Knox Lovell, C. A. K.; Wood, L., "Improving the Allocation of Monetary and Non-Monetary Army Enlistment Incentives by MOS; Validation Efforts Using Quarters Jan-June, 1988, Updating of Parameter Estimates, and Updating of Projections," prepared for U.S. Army Recruiting Command, September 1988.

[26] Pindyck, R. S. and Rubinfeld, D. L. Econometric Models and Economic Forecasts. 2nd edition. New York: McGraw Hill, 1981.

[27] Polich, J. M., and Dertouzos, J. N., "The Enlistment Bonus Experiment," RAND Corporation, WD-2822-FMP, November 1985. (Subsequently published as Polich, J. M.; and Dertouzos, J. N. and Press, J., "The Enlistment Bonus Experiment," RAND Corporation, R-3353-FMP, Santa Monica, 1985.

[28] Studenmund, A. H. and Cassidy, H. J. Using Econometrics A Practical Guide. Boston: Little, Brown and Company, 1987.

[29] Suffa, Fred; Lerro, Pat; Barnes, Jeff, "Resourcing Army Recruiting: Processes and Infrastructure," HumRRO International, Inc., Interim Report 90-03, Alexandria, VA., 1989.

APPENDIX A

Sample RATS Output

```
RATS Version 3.10. 02/01/90
Copyright (c) 1986-90 by VAR Econometrics
Portions (c) 1989-90 by Doan Associates
BMA DATA 15 40 GLOBAL 1000 COM 1000 CON 500 EXP 100 LOCAL 50 $
OPERANDS 100
OUTPUT NOECHO
   T126RM
ARMY 1-3A ENLISTMENT CONTRACTS FOR 12 6'S BY RECRUITING MONTH (SOURCE: USAREC)
   GSMARM
ARMY GSMA CONTRACTS BY RECRUITING MONTH (SOURCE: USAREC)

   T126CM
ARMY 1-3A ENLISTMENT CONTRACTS FOR 12 6'S BY CALENDAR MONTH (SOURCE: USAREC)
   GSMACM
ARMY GSMA CONTRACTS BY CALENDAR MONTH (SOURCE: USAREC)

   DAYSRM
DAYS PER RECRUITING MONTH (SOURCE: USAREC)
   PVWK1821
   PVMILPAY
   UNEMP
ACTUAL AND FORECASTED UNEMPLOYMENT FOR USE IN CONTRACT FORECASTS

   ACFEXPT
EXPECTED VALUE OF ACF BENEFITS -- WEIGHTED AVERAGE ACF BENEFITS TIMES COVERAGE
   ACFCOV
MOS COVERAGE OF ACF BENEFITS -- WEIGHTED AVERAGE @ AVG TOS NPS MALES FY 80-89
   ACFPV
PRESENT VALUE OF ACF BENEFITS -- WEIGHTED AVERAGE @ AVG TOS NPS MALES FY 80-89
   COLLCOST
   BONUS3
   BONCOV3
   AGLM13A
COMBINED SENIORS AND HSDG 1-3A ARMY CONTRACT MISSIONS

   AGLM3B
```

ARMY NPS MALE HSDG CONTRACT MISSIONS: MC = 3B
AROGOAL
ARMY MISSIONS FOR OTHER MALES, FEMALES, AND PRIOR SERVICE (USAREC)

ARECPA
ARMY RECRUITERS (ARECPA IS 66.6% OF ARMYTREC)
TV
NATIONAL TV ADVERTISING EXPENDITURES (SOURCE: USAREC)

RADIO
RADIO ADVERTISING EXPENDITURES (SOURCE: USAREC)
PRINT
PRINT ADVERTISING EXPENDITURES (SOURCE: USAREC)
ADVPDTV
ADVPDRAD
ADVPDMAG
Series CONSTANT (0)
Series RELPAY (9)
Series UNEMP (10)
Series GOALA (21)
Series GOALNOTA (22)
Series ARECPA (26)
Series ADV (27)
Series ACF (11)
Series BON3 (16)
Series BON3SQ (17)
Series BUYUP (35)
Series POL86 (36)
Series DUM688 (38)
Series DUM489 (37)
Series SEAS (34) LAGS -2 TO 0
Series SEAS (34) LAGS -10 TO -7
Series MVG AVGE (-1) LAGS 1 TO 3
Series MVG AVGE (-1) LAGS 7 TO 7
**WARNING 21 Unsatisfactory Initial Estimates. Last MA Set to Zero.
**WARNING 21 Unsatisfactory Initial Estimates. Last MA Set to Zero.
**WARNING 21 Unsatisfactory Initial Estimates. Last MA Set to Zero.
**WARNING 21 Unsatisfactory Initial Estimates. Last MA Set to Zero.

CONVERGENCE REACHED ON ITERATION 41

EQUATION 1
DEPENDENT VARIABLE 4 GSMACM

FROM 80:10 UNTIL 89: 9
TOTAL OBSERVATIONS 108 SKIPPED/MISSING 0
USABLE OBSERVATIONS 108 DEGREES OF FREEDOM 83
R**2 .95106052 RBAR**2 .93690935
SSR .25674379 SEE .55617431E-01
DURBIN-WATSON 1.95326240
Q(26)= 27.6831 SIGNIFICANCE LEVEL .374224

NO.	LABEL	VAR	LAG	COEFFICIENT	STAND. ERROR	T-STATISTIC
***	*******	***	***	**********	**********	**********
1	CONSTANT	0	0	3.461701	2.933158	1.180196
2	RELPAY	9	0	1.160090	.5605583	2.069527
3	UNEMP	10	0	.7619555	.1266181	6.017744
4	GOALA	21	0	.1285634	.5204324E-01	2.470320
5	GOALNOTA	22	0	-.1329527	.4464063E-01	-2.978289
6	ARECPA	26	0	.3468677	.3284425	1.056099
7	ADV	27	0	.2268002E-01	.1357177E-01	1.671117
8	ACF	11	0	.1424108	.4532289E-01	3.142139
9	BON3	16	0	3.695318	4.770987	.7745395
10	BON3SQ	17	0	-64.09608	105.8816	-.6053559
11	BUYUP	35	0	.9944098E-01	.3909786E-01	2.543386
12	POL86	36	0	.1495422	.3344487E-01	4.471303
13	DUM688	38	0	.1402962	.5234005E-01	2.680476
14	DUM489	37	0	-.1952246	.5412922E-01	-3.606640
15	SEAS	34	-2	.1608442	.1922951E-01	8.364448
16	SEAS	34	-1	.7494521E-01	.1932909E-01	3.877327
17	SEAS	34	0	.8019657E-01	.2197501E-01	3.649445
18	SEAS	34	-10	-.1437797	.2288546E-01	-6.282578
19	SEAS	34	-9	.8167688E-01	.2139285E-01	3.817953
20	SEAS	34	-8	.1618502	.1982624E-01	8.163441
21	SEAS	34	-7	.1659229	.2314135E-01	7.169975
22	MVG AVGE	-1	1	.3431384	.1041454	3.294802
23	MVG AVGE	-1	2	.2720845	.1052231	2.585787
24	MVG AVGE	-1	3	.4019551	.1000456	4.017721
25	MVG AVGE	-1	7	-.3685378	.9989340E-01	-3.689311

Goal

56983.00

Forecast

55887.55

REDIN

1.019601

NORMAL COMPLETION OF JOB
HALT AT 0
0 ERRORS 4 WARNINGS

APPENDIX B

REDIN Turbo Pascal Source Code

```
{$R-}    {Range checking off}
{$B+}    {Boolean complete evaluation on}
{$S+}    {Stack checking on}
{$I+}    {I/O checking on}
{$N-}    {No numeric coprocessor}

{$M 6000,0,40000}

{ REDMAIN.PAS - RATS386 Interface Program }

{ Purpose:  redmain.pas is the driver program }

{ Calling Program/Procedure: red.exe  }

{ Written:        07/19/90 : for Jeff Barnes                    }
{ Last Revised:   08/08/90 : by Jack R. Dempsey                 }
{                            Human Resources Research Organization }
{                            (703) 549-3611                     }

program redmain;

{$r-}   {r+ enables rangechecking}
{$v-}   {v- allows passing string parameters of unequal lengths}


Uses
  Crt, {Unit found in TURBO.TPL}
  Dos,
  Printer,
  Turbo3, {Unit found in TURBO3.TPU}
  Winman,
  Windows;

{$i gutildcl.pas}
{$i gutilfwd.pas}
{$i winfw.pas}
{$i int.pas}
{$i gutilcod.pas}
{$i wincd.pas}
```

```
const

  controlpoints =5;
  max_mos = 10;

  valid_eoflds    : allchars = [key_cr,key_esc,key_tab,key_btab,key_uarr,
                               key_darr,key_larr,key_rarr];

  valid_terms     : allchars = [key_cr,key_esc,key_tab,key_btab,key_uarr,
                               key_darr,key_larr,key_rarr];

type
  days        = (Sunday,Monday,Tuesday,Wednesday,Thursday,Friday,Saturday);
  months       = (January,Febrary,March,April,May,June,July,August,September,
                October,November,December);
  c_dw           =array[0..6] of string[9];
  c_mm           =array[1..12] of string[9];

  cntrl_rec    = record
                 cntrl   : array[1..10] of integer;
                 rats_path_sys : string[30];
                 end;


label     166,ginny,lynn,exwife,lilian,barbara,diane,
          martha,debbie2,michelle,michelle1,debbie3,
          denee,18,133,e2,e4,e6,e7,e8,e9,e10,e11,e12,
           19,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12;

const

  dw   : c_dw  = ('Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday');
  mm  : c_mm = ('January','Febrary','March','April','May','June','July','August','September',
                'October','November','December');

var
dys     : array[0..13] of real;

  c_rec          : cntrl_rec;
  cntrldata      : file of cntrl_rec;
  job5data        : text;
  dataset        : text;
  report         : text;
  buf            : string[80];
```

```
job5_data      : string[12];
in_file        : string[30];
old_file       : string[30];
model_file     : string[30];
out_file       : string[30];
rats_path      : string[30];
graph_path     : string[30];
reg_sta_mo     : integer;
reg_sta_yr     : integer;
for_sta_mo     : integer;
for_sta_yr     : integer;
reg_end_mo     : integer;
reg_end_yr     : integer;
for_end_mo     : integer;
for_end_yr     : integer;
time_factor    : integer;
plot_trend     : string[1];
graph_resids   : string[1];
plot_resids    : string[1];
graph_for      : string[1];
plot_for       : string[1];
dtype          : string[1];
nat            : string[1];
dis            : string[1];
step           : integer;
deflator       : string[1];
d_name         : string[8];
graph_run      : string[1];
style          : string[1];
levels         : string[1];
raw            : string[1];
noest          : string[1];
noech          : string[1];
nodet          : string[1];
nocov          : string[1];
nores          : string[1];
nocrs          : string[1];
res3           : array[1..28] of string[8];
inv            : array[1..28] of string[9];
graph          : array[1..28] of string[1];
temp_str       : string[3];
ttype          : string[1];
stats          : string[1];
nent           : real;
tyrs           : integer;
```

```
method          : string[35];
tit             : string[35];
nobs            : real;
maxit           : integer;
depend          : string[8];
dependx         : string[8];
nper            : real;
carg            : real;
page            : integer;
line_no         : integer;
term_option     : char;
choice          : integer;
nvic            : integer;
icurr_sel       : integer;
drive           : char;
drive_sel       : integer;
end_read        : integer;
fy              : real;
col,row         : integer;
sample          : string[1];
trend           : string[1];
win_no          :integer;
win_1           : integer;
win_2           : integer;
i,j,k,m         : integer;
term_char       : char;
cval            : string[1];
rval            : real;
da,yr,mo,dte    : real;
ival            : integer;
fld_no          : integer;
code,nsplits    : integer;
help_scn        : varstring;
finished_entering: boolean;
finished_scn    : boolean;
main_pgm        : file;
pgm_sel         : integer;
set_flag        : integer;
trans_total     : integer;
model           : integer;
oper            : array[1..28] of integer;
trans_in        : array[1..18] of string[70];
trans_out       : array[1..18] of string[70];
flag            : integer;
nvic_rat        : integer;
```

```
  exist          : boolean;
  hour,minute,second,sec100,year,month,day,dayofweek : word;

procedure warn;
begin
  sound(660);
  delay(200);
  nosound;
end;

procedure reset_sys;
begin
 assign(cntrldata,'redcntrl.sys');
 reset(cntrldata);
 with c_rec do
 begin
  cntrl[1]:=0;
  cntrl[2]:=0;
  for j:=3 to 10 do cntrl[j]:=0;
  write(cntrldata,c_rec);
  pgm_sel:=0;
 end;
 close(cntrldata);
end;

procedure options;
begin
  jump(1);
  write(trend);
  jump(2);
  write(graph_resids);
  jump(3);
  write(graph_for);
  jump(4);
  write(dtype);
  jump(5);
  write(stats);
  jump(6);
  write(style);
  jump(7);
end;

procedure estimate;
begin
    reset_screen(singleb,0,white,blue);
```

```
    get_window('exec',win_no);
    set_window(win_no,1,1);
    gotoxy(2,7);
    clreol;
    gotoxy(25,7);
    case model of

1: begin
   write('Writing Rats Source Program');
     swapvectors;
     exec('\command.com','/C redols');
     if(DosError<>0) then
     begin
      Writeln('Dos Error # ',doserror);
      delay(5000);
     end;
     swapvectors;
     gotoxy(2,9);
     clreol;
     gotoxy(25,9);
     write('Unemployment Source Program Complete');
     delay(500);
     gotoxy(2,11);
     clreol;
     gotoxy(25,11);
     write('Loading RATS ');
     gotoxy(2,15);
     clreol;
     gotoxy(32,15);
     Textcolor(30);
     write('Standby .... ');
     textcolor(7);
     swapvectors;
     exec('\command.com','/C '+rats_path+'\rats386 alloc.sys '+out_file);
     if(DosError<>0) then
     begin
      Writeln('Dos Error # ',doserror);
      delay(5000);
     end;
     swapvectors;
    end; {model 1}
2: begin
   write('Writing Rats Source Program');
     swapvectors;
     exec('\command.com','/C redern');
```

```
    if(DosError<>0) then
    begin
     Writeln('Dos Error # ',doserror);
     delay(5000);
    end;
    swapvectors;
    gotoxy(2,9);
    clreol;
    gotoxy(25,9);
    write('Earnings Source Program Complete');
    delay(500);
    gotoxy(2,11);
    clreol;
    gotoxy(25,11);
    write('Loading RATS ');
    gotoxy(2,15);
    clreol;
    gotoxy(32,15);
    Textcolor(30);
    write('Standby .... ');
    textcolor(7);
    swapvectors;
    exec('\command.com','/C '+rats_path+'\rats386 alloc.sys '+out_file);
    if(DosError<>0) then
    begin
     Writeln('Dos Error # ',doserror);
     delay(5000);
    end;
    swapvectors;
   end; {model 2}

3: begin
   write('Writing Rats Source Program');
    swapvectors;
    exec('\command.com','/C redfe');
    if(DosError<>0) then
    begin
     Writeln('Dos Error # ',doserror);
     delay(5000);
    end;
    swapvectors;
    gotoxy(2,9);
    clreol;
    gotoxy(25,9);
    write('Source Program Complete');
```

```
      delay(500);
      gotoxy(2,11);
      clreol;
      gotoxy(25,11);
      write('Loading RATS ');
      gotoxy(2,15);
      clreol;
      gotoxy(32,15);
      Textcolor(30);
      write('Standby .... ');
      textcolor(7);
      swapvectors;
      exec('\command.com','/C '+rats_path+'\rats386 alloc.sys '+out_file);
      if(DosError<>0) then
      begin
       Writeln('Dos Error # ',doserror);
       delay(5000);
      end;
      swapvectors;
     end; {model 3}
    end; {case of model}

end; {procedure estimate}

procedure var_read_data;
label next,exit;

begin
 assign(dataset,'varin.sys');
 rewrite(dataset);
 writeln(dataset,'OPEN DATA ',in_file);
 writeln(dataset,'DEDIT ',in_file);
 writeln(dataset,'CAT');
 writeln(dataset,'END');
 close(dataset);
 swapvectors;
 exec('\command.com','/C '+rats_path+'\rats varin.sys varout.sys');
 if(DosError<>0) then
 begin
  Writeln('Dos Error # ',doserror);
  delay(2000);
 end;
 swapvectors;
 assign(dataset,'varout.sys');;
  {$i-}
```

```
reset(dataset);
  {$i+}
exist:= (ioresult = 0);
flag:=0;
nvic_rat:=0;
j:=0;
if exist then
begin
  while not eof(dataset) do
  begin
   readln(dataset,buf);
   if(copy(buf,2,3)='CAT') then flag:=1;
   if(flag=0) or (copy(buf,2,3)='CAT') then goto next;
   if(copy(buf,2,3)='END') then goto exit;
   if(length(copy(buf,6,8))>0) then
   begin
    res3[1+(6*j)]:=copy(buf,6,8);
    nvic_rat:=nvic_rat+1;
  end;
  if(length(copy(buf,19,8))>0) then
  begin
   res3[2+(6*j)]:=copy(buf,19,8);
   nvic_rat:=nvic_rat+1;
  end;
  if(length(copy(buf,32,8))>0) then
  begin
   res3[3+(6*j)]:=copy(buf,32,8);
   nvic_rat:=nvic_rat+1;
  end;
  if(length(copy(buf,45,8))>0) then
  begin
   res3[4+(6*j)]:=copy(buf,45,8);
   nvic_rat:=nvic_rat+1;
 end;
 if(length(copy(buf,58,8))>0) then
 begin
  res3[5+(6*j)]:=copy(buf,58,8);
  nvic_rat:=nvic_rat+1;
 end;
 if(length(copy(buf,71,8))>0) then
 begin
  res3[6+(6*j)]:=copy(buf,71,8);
  nvic_rat:=nvic_rat+1;
 end;
 j:=j+1;
```

```
    next:
    end;
   exit:
   close(dataset);
   if(nvic_rat>28) then nvic_rat:=28;
   if(nvic<=0) then
   begin
    nvic:=nvic_rat;
    for j:=1 to nvic do
    begin
     inv[j]:='Include  ';
    end;
   end;
  end;
end;

procedure save_red;
label 14;
begin
  gotoxy(2,23);
  clreol;
  gotoxy(32,23);
  textcolor(30);
  write('Saving Data');
  textcolor(7);
  assign(job5data,'red.sys');
  {$i-}
  reset(job5data);
   {$i+}
   exist:= (ioresult = 0);
  if exist then
  begin
   rewrite(job5data);
   writeln(job5data,tit);
   writeln(job5data,method);
   writeln(job5data,in_file);
   writeln(job5data,model_file);
   writeln(job5data,out_file);
   writeln(job5data,rats_path);
   writeln(job5data,graph_path);
   writeln(job5data,nobs:4:0);
   writeln(job5data,nent:2:0);
   writeln(job5data,nper:3:0);
   writeln(job5data,model:1);
   writeln(job5data,ttype);
```

```
      writeln(job5data,reg_sta_mo);
      writeln(job5data,reg_sta_yr);
      writeln(job5data,reg_end_mo);
      writeln(job5data,reg_end_yr);
      writeln(job5data,for_sta_mo);
      writeln(job5data,for_sta_yr);
      writeln(job5data,for_end_mo);
      writeln(job5data,for_end_yr);
      writeln(job5data,nvic);
      writeln(job5data,nvic_rat);
      writeln(job5data,trend);
      writeln(job5data,stats);
      writeln(job5data,graph_resids);
      writeln(job5data,sample);
      writeln(job5data,graph_for);
      writeln(job5data,plot_for);
      writeln(job5data,dtype);
      writeln(job5data,nat);
      writeln(job5data,dis);
      writeln(job5data,graph_run);
      writeln(job5data,levels);
      writeln(job5data,deflator);
      writeln(job5data,d_name);
      writeln(job5data,style);
      writeln(job5data,raw);
      for j:=1 to nvic do
      begin
       if(copy(inv[j],1,1)='C') and (j>nvic_rat) then goto 14;
       writeln(job5data,res3[j]);
       writeln(job5data,inv[j]);
 14:
      end;
      for j:=1 to 18 do
      begin
        writeln(job5data,trans_in[j]);
      end;
      for j:=1 to 18 do
      begin
       writeln(job5data,trans_out[j]);
      end;
      close(job5data);
    end;
  end;

procedure recall_red;
```

```
begin
 assign(job5data,'red.sys');;
 {$i-}
 reset(job5data):
  {$i+}
  exist:= (ioresult = 0);
 if exist then
 begin
  reset(job5data);
  readln(job5data,tit);
  readln(job5data,method);
  readln(job5data,in_file);
  readln(job5data,model_file);
  readln(job5data,out_file);
  readln(job5data,rats_path);
  readln(job5data,graph_path);
  readln(job5data,nobs);
  readln(job5data,nent);
  readln(job5data,nper);
  readln(job5data,model);
  readln(job5data,ttype);
  readln(job5data,reg_sta_mo);
  readln(job5data,reg_sta_yr);
  readln(job5data,reg_end_mo);
  readln(job5data,reg_end_yr);
  readln(job5data,for_sta_mo);
  readln(job5data,for_sta_yr);
  readln(job5data,for_end_mo);
  readln(job5data,for_end_yr);
  readln(job5data,nvic);
  readln(job5data,nvic_rat);
  readln(job5data,trend);
  readln(job5data,stats);
  readln(job5data,graph_resids);
  readln(job5data,sample);
  readln(job5data,graph_for);
  readln(job5data,plot_for);
  readln(job5data,dtype);
  readln(job5data,nat);
  readln(job5data,dis);
  readln(job5data,graph_run);
  readln(job5data,levels);
  readln(job5data,deflator);
  readln(job5data,d_name);
  readln(job5data,style);
```

```
readln(job5data,raw);
for j:=1 to nvic do
begin
 readln(job5data,res3[j]);
 readln(job5data,inv[j]);
end;
for j:=1 to 18 do
begin
  readln(job5data,trans_in[j]);
end;
for j:=1 to 18 do
begin
 readln(job5data,trans_out[j]);
end;
flag:=1;
close(job5data);

end
else
begin
 rewrite(job5data);
 tit:='                                    ';
 method:='      ';
 if(length(old_file)<1) then in_file:='                                ' else
 in_file:=old_file;
 out_file:='                                ';
 model_file:=out_file;
 reg_sta_mo:=10;
 reg_sta_yr:=81;
 reg_end_mo:=9;
 reg_end_yr:=90;
 for_sta_mo:=10;
 for_sta_yr:=81;
 for_end_mo:=9;
 for_end_yr:=90;
 nobs:=5;nent:=1;nper:=5;nvic:=0;trans_total:=18;
 ttype:='A';nvic_rat:=0;
 raw:='Y';
 stats:='N';
 sample:='B';
 plot_trend:='N';
 graph_resids:='N';
 plot_resids:='N';
 graph_for:='N';
 plot_for:='N';
```

```
dtype:='M';
nat:='N';
dis:='N';
trend:='N';
graph_run:='N';
d_name:='None';
deflator:='N';
style:='L';
levels:='Y';
graph_path:='                                    ';
model:=1;
rewrite(job5data);
writeln(job5data,tit);
writeln(job5data,mod);
writeln(job5data,in_file);
writeln(job5data,model_file);
writeln(job5data,out_file);
writeln(job5data,rats_path);
writeln(job5data,graph_path);
writeln(job5data,nobs:4:0);
writeln(job5data,nent:2:0);
writeln(job5data,nper:3:0);
writeln(job5data,model:1);
writeln(job5data,ttype);
writeln(job5data,reg_sta_mo);
writeln(job5data,reg_sta_yr);
writeln(job5data,reg_end_mo);
writeln(job5data,reg_end_yr);
writeln(job5data,for_sta_mo);
writeln(job5data,for_sta_yr);
writeln(job5data,for_end_mo);
writeln(job5data,for_end_yr);
writeln(job5data,nvic);
writeln(job5data,nvic_rat);
writeln(job5data,trend);
writeln(job5data,stats);
writeln(job5data,graph_resids);
writeln(job5data,sample);
writeln(job5data,graph_for);
writeln(job5data,plot_for);
writeln(job5data,dtype);
writeln(job5data,nat);
writeln(job5data,dis);
writeln(job5data,graph_run);
writeln(job5data,levels);
```

```
    writeln(job5data,deflator);
    writeln(job5data,d_name);
    writeln(job5data,style);
    writeln(job5data,raw);
    for j:=1 to 18 do
    begin
     trans_in[j]:=' ';
     writeln(job5data,trans_in[j]);
    end;
    for j:=1 to 18 do
    begin
     trans_out[j]:=' ';
     writeln(job5data,trans_out[j]);
    end;
    close(job5data);
    flag:=0;
   end;
  end;

procedure display_setup;
begin
 jump(1);
 write(tit);
 jump(2);
 write(in_file);
 jump(3);
 write(out_file);
 jump(4);
 write(graph_path);
end;

procedure display_setupred;
begin
 jump(1);
 write(reg_sta_mo:2);
 jump(2);
 write(reg_sta_yr:2);
 jump(3);
 write(reg_end_mo:2);
 jump(4);
 write(reg_end_yr:2);
 jump(5);
 write(for_sta_mo:2);
 jump(6);
 write(for_sta_yr:2);
```

```
 jump(7);
 write(for_end_mo:2);
 jump(8);
 write(for_end_yr:2);
 jump(9);
 write(graph_run);
end;

procedure varinc;

label 11,12,13,14,15,exit;

begin
 for i:=1 to 14 do
   begin
    if((2*(i-1)+1)>nvic) then goto exit;

11: if((2*(i-1)+1)>nvic_rat) and (copy(inv[(2*(i-1))+1],1,1)='C') then
    begin
     for j:=(2*(i-1)+1) to 27 do
     begin
      res3[j]:=res3[j+1];
      inv[j]:=inv[j+1];
      inv[j+1]:='       ';
      res3[j+1]:='      ';
     end;
     nvic:=nvic-1;
     goto 11;
    end;
    if(flag=1) then goto 15;
    jump((4*(i-1))+1);
    if((2*(i-1)+1)<=nvic) and (copy(res3[(2*(i-1))+1],1,1)<>' ') then write(res3[(2*(i-1))+1]);
    jump((4*(i-1))+2);
    if((2*(i-1)+1)<=nvic) and (copy(res3[(2*(i-1))+1],1,1)<>' ') then write(inv[(2*(i-1))+1]);
15: if((2*(i-1)+2)>nvic) then goto exit;
12: if((2*(i-1)+2)>nvic_rat) and (copy(inv[(2*(i-1))+2],1,1)='C') then
    begin
     for j:=(2*(i-1)+2) to 27 do
     begin
      res3[j]:=res3[j+1];
      inv[j]:=inv[j+1];
      inv[j+1]:='       ';
      res3[j+1]:='      ';
     end;
     if((2*(i-1)+2)=28) then
```

```
     begin
      inv[28]:='Clear';
      res3[28]:='       ';
      nvic:=nvic-1;
      goto exit;
     end;
     nvic:=nvic-1;
     goto l2;
    end;
l3:
    if(flag=1) then goto l4;
    jump((4*(i-1))+3);
    if((2*(i-1)+2)<=nvic) and (copy(res3[(2*(i-2))+2],1,1)<>' ') then write(res3[(2*(i-1))+2]);
    jump((4*(i-1))+4);
    if((2*(i-1)+2)<=nvic) and (copy(res3[(2*(i-2))+2],1,1)<>' ') then write(inv[(2*(i-1))+2]);
l4:
   end;
exit: flag:=0;
end;

procedure trans_work;
label try_again,exit_for_good,try,exit;
begin
  k:=1;
  reset_screen(singleb,0,white,blue);
  get_window('trans_in',win_no);
  set_window(win_no,1,1);
  for j:=1 to 18 do
  begin
   jump(k);
  if(length(trans_in[k])<2) then trans_in[k]:=' ';
   write(trans_in[k]);
   k:=k+1;
  end;
  k:=1;
try_again:
  jump(k);
  if(length(trans_in[k])<2) then trans_in[k]:=' ';
  fld_entry(13,ival,rval,trans_in[k],'xxx',valid_eoflds,valid_terms,term_char);
  if(term_char = key_esc) then goto exit;

  if(term_char = key_larr) then
  begin
   k:=k-1;
   if(k<1) then k:=1;
```

```
 goto try_again;
end;

if(term_char = key_rarr) then
begin
 k:=k+1;
 if(k>18) then k:=18;
 goto try_again;
end;

if(term_char = key_uarr) then
begin
 k:=k-1;
 if(k<1) then k:=1;
 goto try_again;
end;

if(term_char = key_darr) then
begin
 k:=k+1;
 if(k>18) then k:=18;
 goto try_again;
end;

trans_total:=18;
k:=k+1;
if(k>18) then goto exit;
goto try_again;

exit:  k:=1;
 reset_screen(singleb,0,white,blue);
 get_window('trans_out',win_no);
 set_window(win_no,1,1);
 for j:=1 to 18 do
 begin
  jump(k);
 if(length(trans_out[k])<2) then trans_out[k]:=' ';
  write(trans_out[k]);
  k:=k+1;
 end;
 k:=1;
try:
 jump(k);
 if(length(trans_out[k])<2) then trans_out[k]:=' ';
 fld_entry(13,ival,rval,trans_out[k],'xxx',valid_eoflds,valid_terms,term_char);
```

```
 if(term_char = key_esc) then goto exit_for_good;

 if(term_char = key_larr) then
 begin
  k:=k-1;
  if(k<1) then k:=1;
  goto try;
 end;

 if(term_char = key_rarr) then
 begin
  k:=k+1;
  if(k>18) then k:=18;
  goto try;
 end;

 if(term_char = key_uarr) then
 begin
  k:=k-1;
  if(k<1) then k:=1;
  goto try;
 end;

 if(term_char = key_darr) then
 begin
  k:=k+1;
  if(k>18) then k:=18;
  goto try;
 end;

 trans_total:=18;
 k:=k+1;
 if(k>18) then goto exit_for_good;
 goto try;
exit_for_good:
end; {procedure trans_work}


begin
   { Display the banner screen. }
   textcolor(7);
   textbackground(1);
   dys[0]:=0;
   dys[1]:=0;
   dys[2]:=31;
```

```
dys[3]:=59;
dys[4]:=90;
dys[5]:=120;
dys[6]:=151;
dys[7]:=181;
dys[8]:=212;
dys[9]:=243;
dys[10]:=273;
dys[11]:=304;
dys[12]:=334;
dys[13]:=365.25;
nvic:=0;
trans_total:=0;
model:=1;
set_cursor (0);
assign(cntrldata,'redcntrl.sys');
{$i-}
reset(cntrldata);
{$i+}
exist:= (ioresult = 0);
if not exist then
begin
 window_init;
 red_int;
 get_window('install',win_no);
 set_window(win_no,1,1);
 assign(cntrldata,'redcntrl.sys');
 rats_path:='                                ';
 in_file:='                                ';
 old_file:=in_file;
 jump(1);
 fld_entry(11,ival,rval,rats_path,'xxx',valid_eoflds,valid_terms,term_char);
 if(term_option = key_esc) then goto 18;
 rewrite(cntrldata);
 with c_rec do
  begin
   cntrl[1]:=0;
   cntrl[2]:=0;
   for j:=3 to 10 do cntrl[j]:=0;
   pgm_sel:=0;
   rats_path_sys:=rats_path;
   write(cntrldata,c_rec);
  end;
close(cntrldata);
end
```

```
else
begin
 assign(cntrldata,'redcntrl.sys');
 reset(cntrldata);
 with c_rec do
 begin
  read(cntrldata,c_rec);
  rats_path:=rats_path_sys;
  pgm_sel:=cntrl[4];
  set_flag:=0;
  if(pgm_sel = 0) then
  begin
   window_init;
   red_int;
  end
  else new_window(80,25,1,1,singleb,60,white,blue,autoz,win_no);
 end;
close(cntrldata);
end;
reset_screen(singleb,0,white,blue);
 get_window('intro',win_no);
 set_window(win_no,1,1);
 set_flag:=0;
 term_option:=' ';
 icurr_sel:=0;
 getdate(year,month,day,dayofweek);
 gettime(hour,minute,second,sec100);
 jump(1);

 if(hour>=0) and (hour<12) then
  begin
   write('Good Morning.');
   jump(2);
   write('Today is ',dw[dayofweek],' ',mm[month],' ',day,', ',year);
  end;
 if(hour>=12) and (hour<18) then
  begin
   write('Good Afternoon.');
   jump(2);
   write('Today is ',dw[dayofweek],' ',mm[month],' ',day,', ',year);
  end;
 if(hour>=18) and (hour<=23) then
  begin
   write('Good Evening.');
   jump(2);
```

```
      write('Today is ',dw[dayofweek],' ',mm[month],' ',day,', ',year);
     end;
    delay(2000);
    icurr_sel:=0;
    set_flag:=0;
ginny:
    reset_screen(singleb,0,white,blue);
    get_window('main',win_no);
    set_window(win_no,1,1);
    for j:=1 to 28 do
    begin
     res3[j]:='      ';
     inv[j]:='Include  ';
    end;

    recall_red;
    old_file:=in_file;
    term_option:=' ';
    icurr_sel:=icurr_sel+1;
    assign(cntrldata,'redcntrl.sys');
    reset(cntrldata);
    with c_rec do
    begin
     read(cntrldata,c_rec);
    end;
    close(cntrldata);
    select_menu_option(icurr_sel,6,'xxx',pgm_sel,term_option);
    if(term_option = key_esc) or (pgm_sel = 6) then goto 18;

    if(flag<1) and (pgm_sel>1) then
    begin
     gotoxy(2,23);
     clreol;
     gotoxy(10,23);
     write('You Must Provide The Data Requested In Setup !!');
     delay(2000);
     pgm_sel:=1;
    end;

19: case pgm_sel of
  1: begin
    reset_screen(singleb,0,white,blue);
    get_window('setup',win_no);
    set_window(win_no,1,1);
    recall_red;
```

```
lynn:
    display_setup;
    gotoxy(2,23);
    clreol;
    gotoxy(25,23);
    write('Enter the Title of the Analysis');
    jump(1);
    if(length(tit)<1) then tit:='                                    ';
    fld_entry(8,ival,rval,tit,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto debbie2;
    if(term_char = key_larr) then goto lynn;
    if(term_char = key_rarr) then goto exwife;
    if(term_char = key_uarr) then goto lynn;
    if(term_char = key_darr) then goto exwife;

exwife:
    gotoxy(2,23);
    clreol;
    gotoxy(19,23);
    write('What is the Path and Name of Your Data File?');
    if(length(in_file)<1) then in_file:='                                    ';
    jump(2);
    fld_entry(11,ival,rval,in_file,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto debbie2;
    if(term_char = key_larr) then goto lynn;
    if(term_char = key_rarr) then goto michelle;
    if(term_char = key_uarr) then goto lynn;
    if(term_char = key_darr) then goto michelle;

michelle:
    gotoxy(2,23);
    clreol;
    gotoxy(19,23);
    write('What is the Path and Name of Your Output File?');
    if(length(out_file)<1) then out_file:='                                    ';
    jump(3);
    fld_entry(11,ival,rval,out_file,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto michelle1;
    if(term_char = key_larr) then goto exwife;
    if(term_char = key_rarr) then goto michelle1;
    if(term_char = key_uarr) then goto exwife;
    if(term_char = key_darr) then goto michelle1;

michelle1:
    gotoxy(2,23);
```

```
    clreol;
    gotoxy(22,23);
    write('What is the Path of Your Graph Output Files?');
    if(length(graph_path)<1) then graph_path:='                              ';
    jump(4);
    fld_entry(11,ival,rval,graph_path,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto debbie2;
    if(term_char = key_larr) then goto michelle;
    if(term_char = key_rarr) then goto debbie2;
    if(term_char = key_uarr) then goto michelle;
    if(term_char = key_darr) then goto debbie2;

debbie2:
    reset_screen(singleb,0,white,blue);
    get_window('model',win_no);
    set_window(win_no,1,1);
    select_menu_option(icurr_sel,3,'xxx',model,term_option);
    if(term_option = key_esc) then goto debbie3;

debbie3:

    jump(5);
    gotoxy(2,23);
    clreol;
    gotoxy(15,23);
    write('Press [S] to Save or [M] to Modify or [esc] to Exit');
    cval:='S';
    gotoxy(68,23);
    fld_entry(4,ival,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto ginny;
    if(cval='M') or (cval='m') then goto 19;
    if(cval<>'S') and (cval<>'s') then goto debbie3;
    flag:=1;
    save_red;

    icurr_sel:=1;
    goto ginny;
  end; {pgm_sel 1}

2: begin
    recall_red;
    reset_screen(singleb,0,white,blue);
    get_window('setupred',win_no);
    set_window(win_no,1,1);
    ttype:='m';
```

```
    time_factor:=12;
    display_setupred;

lilian:
denee:

barbara:
diane:

d1:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Starting Month of the Regression');
    jump(1);
    if(reg_sta_mo<1) or (reg_sta_mo>12) then reg_sta_mo:=10;
    fld_entry(7,reg_sta_mo,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(reg_sta_mo<1) or (reg_sta_mo>12) then goto d1;
    if(term_char = key_larr) then goto denee;
    if(term_char = key_rarr) then goto d2;
    if(term_char = key_uarr) then goto denee;
    if(term_char = key_darr) then goto d2;
d2:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Starting Year of the Regression');
    jump(2);
    if(reg_sta_yr<60) or (reg_sta_yr>99) then reg_sta_yr:=80;
    fld_entry(7,reg_sta_yr,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(reg_sta_yr<60) or (reg_sta_yr>99) then goto d2;
    if(reg_sta_yr<80) then goto d2;
    if(term_char = key_larr) then goto d1;
    if(term_char = key_rarr) then goto d3;
    if(term_char = key_uarr) then goto d1;
    if(term_char = key_darr) then goto d3;
d3:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Ending Month of the Regression');
    jump(3);
    if(reg_end_mo<1) or (reg_end_mo>12) then reg_end_mo:=9;
```

```
    fld_entry(7,reg_end_mo,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(reg_end_mo<1) or (reg_end_mo>12) then goto d3;
    if(term_char = key_larr) then goto d2;
    if(term_char = key_rarr) then goto d4;
    if(term_char = key_uarr) then goto d2;
    if(term_char = key_darr) then goto d4;

d4:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Ending Year of the Regression');
    jump(4);
    if(reg_end_yr<60) or (reg_end_yr>99) then reg_end_yr:=90;
    fld_entry(7,reg_end_yr,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(reg_end_yr<60) or (reg_end_yr>99) then goto d4;
    if(reg_end_yr<reg_sta_yr) then goto d4;
    if(term_char = key_larr) then goto d3;
    if(term_char = key_rarr) then goto d5;
    if(term_char = key_uarr) then goto d3;
    if(term_char = key_darr) then goto d5;
d5:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Starting Month of the Forecast');
    jump(5);
    if(for_sta_mo<1) or (for_sta_mo>12) then for_sta_mo:=10;
    fld_entry(7,for_sta_mo,rval,cval,'xxx',valid_eoflds,valid_terms.term_char);
    if(term_char = key_esc) then goto martha;
    if(for_sta_mo<1) or (for_sta_mo>12) then goto d5;
    if(term_char = key_larr) then goto d4;
    if(term_char = key_rarr) then goto d6;
    if(term_char = key_uarr) then goto d4;
    if(term_char = key_darr) then goto d6;
d6:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Starting Year of the Forecast');
    jump(6);
    if(for_sta_yr<60) or (for_sta_yr>99) then for_sta_yr:=81;
    fld_entry(7,for_sta_yr,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
```

```
    if(term_char = key_esc) then goto martha;
    if(for_sta_yr<60) or (for_sta_yr>99) then goto d6;
    if(for_sta_yr<reg_end_yr) then goto d6;
    if(term_char = key_larr) then goto d5;
    if(term_char = key_rarr) then goto d7;
    if(term_char = key_uarr) then goto d5;
    if(term_char = key_darr) then goto d7;
d7:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Ending Month of the Forecast');
    jump(7);
    if(for_end_mo<1) or (for_end_mo>12) then for_end_mo:=9;
    fld_entry(7,for_end_mo,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(for_end_mo<1) or (for_end_mo>12) then goto d7;
    if(term_char = key_larr) then goto d6;
    if(term_char = key_rarr) then goto d8;
    if(term_char = key_uarr) then goto d6;
    if(term_char = key_darr) then goto d8;

d8:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter the Ending Year of the Forecast');
    jump(8);
    if(for_end_yr<60) or (for_end_yr>99) then for_end_yr:=90;
    fld_entry(7,for_end_yr,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(for_end_yr<60) or (for_end_yr>99) then goto d8;
    if(for_end_yr<for_sta_yr) or (for_end_yr<reg_end_yr) then goto d8;
    if(term_char = key_larr) then goto d7;
    if(term_char = key_rarr) then goto d9;
    if(term_char = key_uarr) then goto d7;
    if(term_char = key_darr) then goto d9;
d9:
d10:
d11:
    gotoxy(2,23);
    clreol;
    gotoxy(19,23);
    write('Is This Run Only To Produce Graphs? (Y/N)');
    jump(9);
```

```
    if(length(graph_run)<1) then graph_run:='N';
    fld_entry(4,ival,rval,graph_run,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto martha;
    if(graph_run<>'N') and (graph_run<>'n') and (graph_run<>'Y') and (graph_run<>'y') then
goto d11;
    if(term_char = key_larr) then goto d8;
    if(term_char = key_rarr) then goto d12;
    if(term_char = key_uarr) then goto d8;
    if(term_char = key_darr) then goto d12;
d12:

martha:
    jump(10);
    gotoxy(2,23);
    clreol;
    gotoxy(15,23);
    write('Press [S] to Save or [M] to Modify or [esc] to Exit');
    cval:='S';
    gotoxy(68,23);
    fld_entry(4,ival,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto ginny;
    if(cval='M') or (cval='m') then goto lilian;
    if(cval<>'S') and (cval<>'s') then goto martha;

    save_red;
    goto ginny;
    end;  {pgm_sel 2}


  3: begin {Plots/Graphs}
     reset_screen(singleb,0,white,blue);
     get_window('options',win_no);
     set_window(win_no,1,1);
     goto 133;
166:

    gotoxy(2,23);
    clreol;
    gotoxy(29,23);
    write('Enter Yes or No (Y/N)');
    jump(1);
    fld_entry(4,ival,rval,trend,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto 133;
    if(trend<>'Y') and (trend<>'y') and (trend<>'N') and (trend<>'n') then goto 166;
    if(term_char = key_larr) then goto 166;
```

```
    if(term_char = key_rarr) then goto e2;
    if(term_char = key_uarr) then goto 166;
    if(term_char = key_darr) then goto e2;

e2:

    gotoxy(2,23);
    clreol;
    gotoxy(29,23);
    write('Enter Yes or No (Y/N)');
    jump(2);
    fld_entry(4,ival,rval,graph_resids,'xxx',valid_eoflds,valid_terms,term_char);
    graph_resids:='N';
    if(term_char = key_esc) then goto 133;
    if(graph_resids<>'Y') and (graph_resids<>'y') and (graph_resids<>'N') and
(graph_resids<>'n') then goto e2;
    if(term_char = key_larr) then goto e2;
    if(term_char = key_rarr) then goto e4;
    if(term_char = key_uarr) then goto e2;
    if(term_char = key_darr) then goto e4;

e4:

    gotoxy(2,23);
    clreol;
    gotoxy(29,23);
    write('Enter Yes or No (Y/N)');
    jump(3);
    fld_entry(4,ival,rval,graph_for,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto 133;
    if(graph_for<>'Y') and (graph_for<>'y') and (graph_for<>'N') and (graph_for<>'n') then
goto e4;
    if(term_char = key_larr) then goto e4;
    if(term_char = key_rarr) then goto e6;
    if(term_char = key_uarr) then goto e4;
    if(term_char = key_darr) then goto e6;

e6:
    gotoxy(2,23);
    clreol;
    gotoxy(9,23);
    write('Enter whether To Present Monthly, Quarterly or Annual');
    jump(4);
    if(dtype<>'M') and (dtype<>'m') and (dtype<>'Q') and (dtype<>'q') and
    (dtype<>'A') and (dtype<>'a') then dtype:='M';
```

```
fld_entry(4,ival,rval,dtype,'xxx',valid_eoflds,valid_terms,term_char);
dtype:='M';
if(term_char = key_esc) then goto 133;
if(dtype<>'M') and (dtype<>'m') and (dtype<>'Q') and (dtype<>'q') and
(dtype<>'A') and (dtype<>'a') then goto e6;
if((dtype='M') or (dtype='m')) and ((ttype='Q') or (ttype='q')) then
begin
 warn;
 gotoxy(2,23);
 clreol;
 textcolor(30);
 gotoxy(9,23);
 write('You Cannot Obtain Monthly Graphs Using Quarterly Data');
 textcolor(7);
 delay(1000);
 goto e6;
end;
if((dtype='M') or (dtype='m')) and ((ttype='A') or (ttype='a')) then
begin
 warn;
 gotoxy(2,23);
 clreol;
 textcolor(30);
 gotoxy(9,23);
 write('You Cannot Obtain Monthly Graphs Using Annual Data');
 textcolor(7);
 delay(1000);
 goto e6;
end;
if((dtype='Q') or (dtype='q')) and ((ttype='A') or (ttype='a')) then
begin
 warn;
 gotoxy(2,23);
 clreol;
 textcolor(30);
 gotoxy(9,23);
 write('You Cannot Obtain Quarterly Graphs Using Annual Data');
 textcolor(7);
 delay(1000);
 goto e6;
end;

if(term_char = key_larr) then goto e4;
if(term_char = key_rarr) then goto e7;
if(term_char = key_uarr) then goto e4;
```

```
    if(term_char = key_darr) then goto e7;

e7:
   nat:='Y';
e8:

e9:

e10:

    gotoxy(2,23);
    clreol;
    gotoxy(29,23);
    write('Enter Yes or No (Y/N)');
    jump(5);
    if(length(stats)<1) then stats:='N';
    fld_entry(4,ival,rval,stats,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto l33;
    if(stats<>'Y') and (stats<>'y') and (stats<>'N') and (stats<>'n') then goto e10;
    if(term_char = key_larr) then goto e6;
    if(term_char = key_rarr) then goto e11;
    if(term_char = key_uarr) then goto e6;
    if(term_char = key_darr) then goto e11;
e11:

    gotoxy(2,23);
    clreol;
    gotoxy(29,23);
    write('Enter Yes or No (Y/N)');
    jump(6);
    if(length(style)<1) then stats:='L';
    fld_entry(4,ival,rval,style,'xxx',valid_eoflds,valid_terms,term_char);
    if(term_char = key_esc) then goto l33;
    if(style<>'L') and (style<>'l') and (style<>'B') and (style<>'b')
     and (style<>'P') and (style<>'p') then goto e11;
    if(term_char = key_larr) then goto e10;
    if(term_char = key_rarr) then goto e12;
    if(term_char = key_uarr) then goto e10;
    if(term_char = key_darr) then goto e12;
e12:


l33: jump(7);
    options;
    gotoxy(2,23);
```

```
  clreol;
  gotoxy(15,23);
  write('Press [S] to Save or [M] to Modify or [esc] to Exit');
  cval:='S';
  gotoxy(68,23);
  fld_entry(4,ival,rval,cval,'xxx',valid_eoflds,valid_terms,term_char);
  if(term_char = key_esc) then goto ginny;
  if(cval='m') or (cval='M') then goto 166;
  if(cval<>'S') and (cval<>'s') then goto 133;
  save_red;
  goto ginny;
 end; {pgm_sel 4}

 4: begin    {Execute Rats}
    estimate;
    goto ginny;
   end; {pgm_sel 5}

 5: begin {DATA Editor}
    swapvectors;
    exec('\command.com','/C '+rats_path+'\ratsdata');
    if(DosError<>0) then
    begin
     Writeln('Dos Error # ',doserror);
     delay(5000);
    end;
    swapvectors;
    icurr_sel:=3;
    goto ginny;
   end; {pgm_sel 6}

 end;   {end of pgm_sel}

18:
   assign(cntrldata,'redcntrl.sys');
   reset(cntrldata);
   with c_rec do
   begin
    cntrl[1]:=0;
    cntrl[2]:=0;
    rats_path_sys:=rats_path;
    for j:=4 to 10 do cntrl[j]:=0;
    write(cntrldata,c_rec);
   end;
   close(cntrldata);
```

```
  reset_screen(noneb,0,white,blue);
  set_cursor(2);
  halt;
  end. {end of redmain}



{$R-}    {Range checking off}
{$B+}    {Boolean complete evaluation on}
{$S+}    {Stack checking on}
{$I+}    {I/O checking on}
{$N-}    {No numeric coprocessor}

{$M 6000,0,40000}

{ redern.PAS - RATS386 Interface Program }

{ Purpose:  redmain.pas is the driver program }

{ Calling Program/Procedure: redmain.exe  }

{ Written:        08/08/90 : For Jeff Barnes              }
{ Last Revised:   08/11/90 : by Jack R. Dempsey           }

program redern;

{$r-}   {r+ enables rangechecking}
{$v-}   {v- allows passing string parameters of unequal lengths}


Uses
  crt,
  Dos;

{$i gutildcl.pas}
{$i erlfefwd.pas}
{$i erlfecod.pas}

var
  job5data       : text;
  uataset        : text;
  report         : text;
  buf            : string[80];
  job5_data      : string[12];
  in_file        : string[30];
```

```
model_file      : string[30];
out_file        : string[30];
rats_path       : string[30];
graph_path      : string[30];
reg_sta_mo      : integer;
reg_sta_yr      : integer;
for_sta_mo      : integer;
for_sta_yr      : integer;
reg_end_mo      : integer;
reg_end_yr      : integer;
for_end_mo      : integer;
for_end_yr      : integer;
time_factor     : integer;
stats           : string[1];
graph_resids    : string[1];
sample          : string[1];
graph_for       : string[1];
plot_for        : string[1];
dtype           : string[1];
nat             : string[1];
dis             : string[1];
deflator        : string[1];
d_name          : string[8];
graph_run       : string[1];
levels          : string[1];
style           : string[1];
raw             : string[1];
step            : integer;
noest           : string[1];
noech           : string[1];
nodet           : string[1];
nocov           : string[1];
model           : integer;
nores           : string[1];
nocrs           : string[1];
res3            : array[1..28] of string[8];
inv             : array[1..28] of string[9];
temp_str        : string[3];
ttype           : string[1];
nent            : integer;
tyrs            : integer;
method          : string[35];
tit             : string[35];
nobs            : real;
maxit           : integer;
```

```
  depend          : string[8];
  dependx         : string[8];
  nper            : real;
  carg            : real;
  page            : integer;
  line_no         : integer;
  term_option     : char;
  choice          : integer;
  nvic            : integer;
  icurr_sel       : integer;
  drive           : char;
  drive_sel       : integer;
  end_read        : integer;
  fy              : real;
  col,row         : integer;
  trend           : string[1];
  i,j,k,m         : integer;
  code,nsplits    : integer;
  finished_entering: boolean;
  finished_scn    : boolean;
  main_pgm        : file;
  pgm_sel         : integer;
  set_flag        : integer;
  trans_total     : integer;
  model_total     : integer;
  oper            : array[1..28] of integer;
  trans_in        : array[1..18] of string[70];
  PP              : PathStr;
  DD              : DirStr;
  NN              : NameStr;
  EE              : ExtStr;
  flag            : integer;
  nvic_rat        : integer;
  exist           : boolean;


procedure open_data;

begin

 assign(dataset,'alloc.sys');
 rewrite(dataset);
 writeln(dataset,'*****************************************************************
***********');
 writeln(dataset,'*  Interface by Jack R. Dempsey Human Resources Research Organization *');
```

```
writeln(dataset,'*  Questions call (703) 549-3611                                    *');
writeln(dataset,'*************************************************************************
**********');
writeln(dataset,'output noecho');
PP:=in_file;
fsplit(PP,DD,NN,EE);
writeln(dataset,' ');
writeln(dataset,'open data dates.sys');
writeln(dataset,'declare vector dates(10)');
writeln(dataset,'input(UNIT=DATA) dates');
writeln(dataset,' ');
writeln(dataset,'cal FIX(dates(1)) FIX(dates(2)) 4 ;*QUARTERLY CALENDAR');
writeln(dataset,' ');
writeln(dataset,'ieval dlastfc = cal(fix(dates(9)),fix(dates(10)))');
writeln(dataset,'  WRITE DLASTFC');
writeln(dataset,'all 0 dlastfc+4');
writeln(dataset,' ');
writeln(dataset,'IEVAL DLAST = dlastfc-6');
writeln(dataset,'ieval nfor = dlastfc - dlast');
writeln(dataset,'CLEAR UNEMP CPI WE1624 WE1619 WE2024');
writeln(dataset,'open data ',in_file,'; * MONTHLY SOURCE DATABASE');
writeln(dataset,'data(format=rats)  78,2 DLASTFC unemp');
writeln(dataset,'open data ',DD,'ewsannex.rat ; * QUARTERLY SOURCE DATABASE');
writeln(dataset,'data(format=rats)  78,3 DLASTFC cpi');
writeln(dataset,'data(format=rats)  79,1 DLAST we1624 we1619 we2024');
writeln(dataset,'');
writeln(dataset,' ');
writeln(dataset,'set trend 77,3 dlastfc = t');
writeln(dataset,'set trend2 77,3 dlastfc = t * t');
writeln(dataset,' ');
writeln(dataset,' ');
writeln(dataset,'clear seas');
writeln(dataset,'seasonal seas 77,3 dlastfc+2 4 78,1');
writeln(dataset,' ');
writeln(dataset,'* projecting civ. earnings');
writeln(dataset,'CLEAR EGROW CGROW LFCAST');
writeln(dataset,'set egrow 80,1 dlast = (we2024(t)/we2024(t-4)-1)*100.');
writeln(dataset,'set cgrow 79,3 dlastfc = (cpi(t)/cpi(t-4)-1)*100.');
writeln(dataset,'equation(more) 1 egrow');
writeln(dataset,'# constant  unemp -cgrow 1 1');
writeln(dataset,'initial 1');
writeln(dataset,'clear resids');
writeln(dataset,'iterate 1 80,1 DLAST resids');
writeln(dataset,'set earnhat 79,1 DLAST = egrow(t) - resids(t)');
writeln(dataset,'set actual 79,1 DLAST = egrow(t)');
```

```
writeln(dataset,'set error 79,1 DLAST = earnhat(t) - actual(t)');
writeln(dataset,'print(dates) 79,1 DLAST we2024 actual earnhat error');
writeln(dataset,'forecast(print) 1 6 dlast+1');
writeln(dataset,'# 1 lfcast dlast+1');
writeln(dataset,'set egrow dlast+1 DLASTFC = lfcast(t)');
writeln(dataset,'set we2024 dlast+1 dlastfc = (1.0 + egrow(t)/100.) * we2024(t-4)');
writeln(dataset,'set lfcast dlast dlast = 0.0');
writeln(dataset,'print(dates) dlast dlastfc lfcast egrow we2024');
writeln(dataset,' ');
writeln(dataset,'* >>>>>creating four year stream & calculating present value');
writeln(dataset,'clear pvcivwe grate');
writeln(dataset,'set grate 79,1 dlast = (we2024(t)/we1619(t))**0.25');
writeln(dataset,'set grate dlast+1 dlastfc = grate(dlast)');
writeln(dataset,'eval discf = 1.3');
writeln(dataset,'set we1619 dlast+1 dlastfc = we2024(t)/grate(t)**4');
writeln(dataset,'set pvcivwe 79,1 dlastfc = we1619(t)*(1+(grate(t)/discf)+(grate(t)/discf)**2 $');
writeln(dataset,'  +(grate(t)/discf)**3)');
writeln(dataset,'print(dates) 79,1 dlastfc we1619 we2024 grate pvcivwe');
writeln(dataset,' ');
writeln(dataset,'* proceed to deseasonalization');
writeln(dataset,'clear resids www fitted');
writeln(dataset,'LINREG pvcivwe 79,1 DLAST resids www');
writeln(dataset,'# constant trend -seas 0 -2');
writeln(dataset,'prj fitted 79,1 dlastfc');
writeln(dataset,'set resids dlast+1 dlastfc = pvcivwe(t) - fitted(t)');
writeln(dataset,'print(dates) 79,1 dlastfc pvcivwe fitted resids');
writeln(dataset,' ');
writeln(dataset,'statistics pvcivwe 79,1 DLASTFC');
writeln(dataset,'set ci 79,1 DLASTFC = resids(t) + mean');
writeln(dataset,' ');
writeln(dataset,'set ztime 79,1 DLASTFC = www(2) * trend(t)');
writeln(dataset,'statistics ztime 79,1 DLASTFC');
writeln(dataset,'set cit 79,1 DLASTFC = ci(t) + ztime(t) - mean');
writeln(dataset,' ');
writeln(dataset,'statistics pvcivwe 79,1 DLASTFC');
writeln(dataset,'statistics cit 79,1 DLASTFC');
writeln(dataset,'print(dates) 79,1 DLASTFC pvcivwe cit ci');
writeln(dataset,' ');
writeln(dataset,'DEDIT ',DD,'EWSANNEX.RAT');
writeln(dataset,'dreplace WECIT CIT 79,1 DLASTFC 1');
writeln(dataset,'  QUARTERLY, SEASONALLY-ADJUSTED PAY VARIABLE TO BE
INTERPOLATED MONTHLY.');
writeln(dataset,'PRTDATA WECIT');
writeln(dataset,'SAVE');
writeln(dataset,' ');
```

```
 writeln(dataset,'* save the "cit" series and convert it into a monthly series');
 writeln(dataset,'END 1');
 writeln(dataset,'* CONVERSION TO A MONTHLY FREQUENCY');
 writeln(dataset,' ');
 writeln(dataset,'open data dates.sys');
 writeln(dataset,'declare vector dates(10)');
 writeln(dataset,'input(UNIT=DATA) dates');
 writeln(dataset,' ');
 writeln(dataset,'cal FIX(dates(1)) FIX(dates(2))  FIX(dates(3))');
 writeln(dataset,'IEVAL DLSTEFC = FIX(DATES(5)) + 2');
 writeln(dataset,'ieval dlastfc = cal(fix(dates(4)),dLstefc)');
 writeln(dataset,'all 0 dlastfc+10 ; * last month in a quarter should be used here');
 writeln(dataset,'CLEAR WECIT');
 writeln(dataset,'open data ',DD,'ewsannex.rat');
 writeln(dataset,'data(format=rats) 79,1 dlastfc wecit');
 writeln(dataset,'ieval dfirst = (79,1) ; ieval dlastm2 = dlastfc - 2');
 writeln(dataset,'clear mx');
 writeln(dataset,'set(scratch) mx dfirst dfirst = wecit(t)');
 writeln(dataset,'ieval dfirst = dfirst + 1');
 writeln(dataset,'do i = dfirst,dlastm2,3');
 writeln(dataset,'   set(scratch) mx i i = wecit(t)');
 writeln(dataset,'   ieval d2nd = i + 1 ; ieval d3rd = i + 2');
 writeln(dataset,'   set(scratch) mx d2nd d2nd = wecit(t-1)+(wecit(t+2) - wecit(t-1))/3');
 writeln(dataset,'   set(scratch) mx d3rd d3rd = mx(t-1)+(wecit(t+1) - wecit(t-2))/3');
 writeln(dataset,'end do i');
 writeln(dataset,'set mx dlastm2+1 dlastfc = wecit(t)');
 writeln(dataset,'dedit ',in_file);
 writeln(dataset,'prtdata pvwk1821');
 writeln(dataset,'dreplace pvwk1821 mx 79,1 dlastfc');
 writeln(dataset,'prtdata pvwk1821');
 writeln(dataset,'save');
 writeln(dataset,'end');
 close(dataset);
end; {open_data}

procedure mdates;
begin
 assign(dataset,'dates.sys');
 rewrite(dataset);
 writeln(dataset,' 69  1  12  ',for_end_yr,'  ',for_end_mo,'  86  10  0  ',for_end_yr,'  4');
 writeln(dataset,'* ^   ^  ^   ^   ^  ^   ^   ^  ^   ^');
 writeln(dataset,'* B   B  F   D   D  F   F     D   D');
 writeln(dataset,'* A   A  R   L   L  I   I     L   L');
 writeln(dataset,'* S   S  E   A   A  R   R     A   A');
 writeln(dataset,'* E   E  Q   S   S  S   S     S   S');
```

```
 writeln(dataset,'*          T  T T  T    T  T');
 writeln(dataset,'* Y  M     F  F         F  F');
 writeln(dataset,'* R  O     C  C F  F    C  C');
 writeln(dataset,'*             Y  Y');
 writeln(dataset,'*          Y  M         Q  Q');
 writeln(dataset,'*          R  O Y  M    Y  T');
 writeln(dataset,'*             R  O      R  R');
 close(dataset);
end; {mdates}

procedure recall_red;
begin
  assign(job5data,'red.sys');;
  {$i-}
  reset(job5data);
   {$i+}
   exist:= (ioresult = 0);
  if exist then
  begin
   reset(job5data);
   readln(job5data,tit);
   readln(job5data,method);
   readln(job5data,in_file);
   readln(job5data,model_file);
   readln(job5data,out_file);
   readln(job5data,rats_path);
   readln(job5data,graph_path);
   readln(job5data,nobs);
   readln(job5data,nent);
   readln(job5data,nper);
   readln(job5data,model);
   readln(job5data,ttype);
   readln(job5data,reg_sta_mo);
   readln(job5data,reg_sta_yr);
   readln(job5data,reg_end_mo);
   readln(job5data,reg_end_yr);
   readln(job5data,for_sta_mo);
   readln(job5data,for_sta_yr);
   readln(job5data,for_end_mo);
   readln(job5data,for_end_yr);
   readln(job5data,nvic);
   readln(job5data,nvic_rat);
   readln(job5data,trend);
   readln(job5data,stats);
   readln(job5data,graph_resids);
```

```
  readln(job5data,sample);
  readln(job5data,graph_for);
  readln(job5data,plot_for);
  readln(job5data,dtype);
  readln(job5data,nat);
  readln(job5data,dis);
  readln(job5data,graph_run);
  readln(job5data,levels);
  readln(job5data,deflator);
  readln(job5data,d_name);
  readln(job5data,style);
  readln(job5data,raw);
  for j:=1 to nvic do
  begin
   readln(job5data,res3[j]);
   readln(job5data,inv[j]);
  end;
  for j:=1 to 18 do
  begin
    readln(job5data,trans_in[j]);
  end;
  flag:=1;
  close(job5data);
 end;
end;

begin
recall_red;
mdates;
open_data;
end. {end of redern}


{$R-}    {Range checking off}
{$B+}    {Boolean complete evaluation on}
{$S+}    {Stack checking on}
{$I+}    {I/O checking on}
{$N-}    {No numeric coprocessor}

{$M 6000,0,40000}

{ red01.PAS - RATS386 Interface Program }

{ Purpose:  redmain.pas is the driver program }
```

```
{ Calling Program/Procedure: redmain.exe  }

{ Written:        08/08/90 : for Jeff Barnes              }
{ Last Revised:   08/10/90 : by Jack R. Dempsey           }

program red01;

{$r-}   {r+ enables rangechecking}
{$v-}   {v- allows passing string parameters of unequal lengths}


Uses
  crt,
  Dos;

{$i gutildcl.pas}
{$i erlfefwd.pas}
{$i erlfecod.pas}

var
  job5data        : text;
  dataset         : text;
  report          : text;
  buf             : string[80];
  job5_data       : string[12];
  in_file         : string[30];
  model_file      : string[30];
  out_file        : string[30];
  rats_path       : string[30];
  graph_path      : string[30];
  reg_sta_mo      : integer;
  reg_sta_yr      : integer;
  for_sta_mo      : integer;
  for_sta_yr      : integer;
  reg_end_mo      : integer;
  reg_end_yr      : integer;
  for_end_mo      : integer;
  for_end_yr      : integer;
  time_factor     : integer;
  stats           : string[1];
  graph_resids    : string[1];
  sample          : string[1];
  graph_for       : string[1];
  plot_for        : string[1];
  dtype           : string[1];
```

```
nat             : string[1];
dis             : string[1],
deflator        : string[1];
d_name          : string[8];
graph_run       : string[1];
levels          : string[1];
style           : string[1];
raw             : string[1];
step            : integer;
noest           : string[1];
noech           : string[1];
noset           : string[1];
nocov           : string[1];
model           : integer;
nores           : string[1];
nocrs           : string[1];
res3            : array[1..28] of string[8];
inv             : array[1..28] of string[9];
temp_str        : string[3];
ttype           : string[1];
nent            : integer;
tyrs            : integer;
method          : string[35];
tit             : string[35];
nobs            : real;
maxit           : integer;
depend          : string[8];
dependx         : string[8];
nper            : real;
carg            : real;
page            : integer;
line_no         : integer;
term_option     : char;
choice          : integer;
nvic            : integer;
icurr_sel       : integer;
drive           : char;
drive_sel       : integer;
end_read        : integer;
fy              : real;
col,row         : integer;
trend           : string[1];
i,j,k,m         : integer;
code,nsplits    : integer;
finished_entering: boolean;
```

```
  finished_scn     : boolean;
  main_pgm         : file;
  pgm_sel          : integer;
  set_flag         : integer;
  trans_total      : integer;
  model_total      : integer;
  oper             : array[1..28] of integer;
  trans_in         : array[1..18] of string[70];
  flag             : integer;
  nvic_rat         : integer;
  exist            : boolean;


procedure stats_out;

var
kk,jj    : integer;
adj      : integer;
stile1   : string[10];
stile2   : string[10];
mm       : integer;

begin
 mm:=nent;
 stile1:='LINE';
 stile2:='LINE';
 if(style='b') or (style='B') then stile1:='BARGRAPH';
 if(style='b') or (style='B') then stile2:='OVERLAP';
 if(style='p') or (style='P') then stile1:='POLYGONAL';
 if(raw='y') or (raw='Y') then kk:=nvic_rat else kk:=nvic;

 if((trend='Y') or (trend='y')) and ((dis='Y') or (dis='y'))
 and ((dtype='M') or (dtype='m')) and (time_factor>4) then
 begin
  for j:=1 to kk do
  begin
   if(copy(inv[j],1,1)<>'C') then
   begin
    right_trim(graph_path);
    right_trim(res3[j]);
    for jj:=1 to mm do
    begin
     if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,res3[j],'.GSP');
     writeln(dataset,' SET VAR1 1 MO = ',res3[j],'((t-1)*BLK+1+(',jj,'-1))');
     writeln(dataset,' DISPLAY(STORE=HEADER) ''DISTRICT ',jj,' TRENDS (Monthly)''');
```

```
    writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Months'', $');
    writeln(dataset,'STYLE=',stile1,',VLABEL=''',res3[j],''')');
    writeln(dataset,' # VAR1');
    writeln(dataset,'CLEAR VAR1');
    if(length(graph_path)>1) then writeln(dataset,'CLOSE PLOT');
   end;
  end;
 end;
end;

 if((trend='Y') or (trend='y')) and ((dis='Y') or (dis='y'))
 and ((dtype='Q') or (dtype='q')) and (time_factor>=4) then
 begin
  adj:=1;
  if(ttype='M') or (ttype='m') then adj:=3;
  for j:=1 to kk do
  begin
   if(copy(inv[j],1,1)<>'C') then
   begin
   for jj:=1 to mm do
   begin
    if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\',res3[j],'.GSP');
    writeln(dataset,' SET VAR1 / = ',res3[j],'((t-1)*BLK+1+(',jj,'-1))');
    writeln(dataset,'DO J=1,MO/',adj);
    writeln(dataset,' ACCUMULATE VAR1 ((J-1)*',adj,')+1 (J*',adj,') SUM1 ((J-1)*',adj,')+1');
    writeln(dataset,'END DO J');
    writeln(dataset,' SET VAR2 1 MO/',adj,' = SUM1(t*',adj,')');
    writeln(dataset,' CLEAR SUM1');
    writeln(dataset,' DISPLAY(STORE=HEADER) ''DISTRICT ',jj,' TRENDS (Quarterly)''');
    writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Quarters'', $');
    writeln(dataset,'STYLE=',stile1,',VLABEL=''',res3[j],''')');
    writeln(dataset,' # VAR2');
   end;
  end;
 end;
 writeln(dataset,'CLEAR VAR1 VAR2');
end;

 if((trend='Y') or (trend='y')) and ((dis='Y') or (dis='y'))
 and ((dtype='A') or (dtype='a')) and (time_factor>=1) then
 begin
  adj:=1;
  if(ttype='M') or (ttype='m') then adj:=12;
  if(ttype='Q') or (ttype='q') then adj:=3;
  for j:=1 to kk do
```

```
 begin
  if(copy(inv[j],1,1)<>'C') then
  begin
  for jj:=1 to mm do
  begin
   if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\',res3[j],'.GSP');
   writeln(dataset,' SET VAR1 / = ',res3[j],'((t-1)*BLK+1+(',jj,'-1))');
   writeln(dataset,'DO J=1,MO/',adj);
   writeln(dataset,' ACCUMULATE VAR1 ((J-1)*',adj,')+1 (J*',adj,') SUM1 ((J-1)*',adj,')+1');
   writeln(dataset,'END DO J');
   writeln(dataset,' SET VAR2 1 MO/',adj,' = SUM1(t*',adj,')');
   writeln(dataset,' CLEAR SUM1 VAR1');
   writeln(dataset,' DISPLAY(STORE=HEADER) ''DISTRICT ',jj,' TRENDS (Annual)''');
   writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Year'', $');
   writeln(dataset,'STYLE=',stile1,',VLABEL=''',res3[j],''')');
   writeln(dataset,' # VAR2');
  end;
 end;
end;
writeln(dataset,'CLEAR VAR2');
end;

dtype:='m';
if((trend='Y') or (trend='y')) and ((nat='Y') or (nat='y'))
and ((dtype= M') or (dtype='m')) and (time_factor>4) then
begin
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL TRENDS''');
 for j:=1 to kk do
 begin
  if(copy(inv[j],1,1)<>'C') then
  begin
   if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\',res3[j],'.GSP');
   writeln(dataset,'GRAPH(HEADER=HEADER, $');
   writeln(dataset,'STYLE=',stile1,',VLABEL=''',res3[j],''')');
   writeln(dataset,' # ',res3[j]);
  end;
 end;
end;

if((trend='Y') or (trend='y')) and ((nat='Y') or (nat='y'))
and ((dtype='Q') or (dtype='q')) and (time_factor>=4) then
begin
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL TRENDS (Quarterly)''');
 adj:=1;
 if(ttype='M') or (ttype='m') then
```

```
 begin
  adj:=3;
 end;
 for j:=1 to kk do
 begin
  if(copy(inv[j],1,1)<>'C') then
  begin
   if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\',res3[j],'.GSP');
   writeln(dataset,'DO I=1,MO/',adj);
   writeln(dataset,' ACCUMULATE ',res3[j],' ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
   writeln(dataset,'END DO I');
   writeln(dataset,' SET VAR1 1 MO/',adj,' = SUM1(t*BLK*',adj,')/BLK');
   writeln(dataset,' CLEAR SUM1');
   writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Quarter'', $');
   writeln(dataset,'STYLE=',stile1,',VLABEL=''',res3[j],''')');
   writeln(dataset,' # VAR1');
   writeln(dataset,'CLEAR VAR1');

  end;
 end;
end;
if((trend='Y') or (trend='y')) and ((nat='Y') or (nat='y'))
and ((dtype='A') or (dtype='a')) and (time_factor>=1) then
begin
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL TRENDS (Annual)''');
 adj:=1;
 if(ttype='M') or (ttype='m') then adj:=12;
 if(ttype='Q') or (ttype='q') then adj:=3;
 for j:=1 to kk do
 begin
  if(copy(inv[j],1,1)<>'C') then
  begin
   if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\',res3[j],'.GSP');
   writeln(dataset,'DO I=1,MO/',adj);
   writeln(dataset,' ACCUMULATE ',res3[j],' ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
   writeln(dataset,'END DO I');
   writeln(dataset,' SET VAR1 1 MO/',adj,' = SUM1(t*BLK*',adj,')/BLK');
   writeln(dataset,' CLEAR SUM1');
   writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Year'', $');
   writeln(dataset,'STYLE=',stile1,',VLABEL=''',res3[j],''')');
   writeln(dataset,' # VAR1');
   writeln(dataset,'CLEAR VAR1');
```

```
  end;
 end;
end;
end; {stats_out}


procedure national;
var adj    : integer;
stile1   : string[10];
stile2   : string[10];

begin
 stile1:='LINE';
 stile2:='LINE';
 if(style='b') or (style='B') then stile1:='BARGRAPH';
 if(style='b') or (style='B') then stile2:='OVERLAP';
 if(style='p') or (style='P') then stile1:='POLYGONAL';

 if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
 and ((dtype='M') or (dtype='m')) then
 begin
  right_trim(graph_path);
  if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NLM.GSP');
  writeln(dataset,'DO I=1,MO');
  writeln(dataset,' ACCUMUL\TE levelprj ((I-1)*BLK)+1 (I*BLK) SUM1 ((I-1)*BLK)+1');
  writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK)+1 (I*BLK) SUM2 ((I-1)*BLK)+1');
  writeln(dataset,'END DO I');
  writeln(dataset,' SET VAR1 ',time_factor,'+1 MO = SUM1(t*BLK)');
  writeln(dataset,  SET VAR2 ',time_factor,'+1 MO = SUM2(t*BLK)');
  writeln(dataset,' CLEAR SUM1 SUM2');
  writeln(dataset,'SPGRAPH(TWOSCALE)');
  writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL ACTUAL vs. FORECAST ''');
  writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Months'', $');
  writeln(dataset,'STYLE=',stile2,',VLABEL=''Forecast vs. Actual'')');
  writeln(dataset,' # VAR1');
  writeln(dataset,'GRAPH 1');
  writeln(dataset,' # VAR2');
  writeln(dataset,'SPGRAPH(DONE)');
 end;
 writeln(dataset,' ');

 if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
 and ((dtype='Q') or (dtype='q')) and (time_factor>=4) then
 begin
  if(ttype='M') or (ttype='m') then adj:=3;
```

```
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NLQ.GSP');
 writeln(dataset,'DO I=1,MO/',adj);
 writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
 writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
 writeln(dataset,'END DO I');
 writeln(dataset,' SET VAR1 1 MO/',adj,' = SUM1(t*BLK*',adj,')');
 writeln(dataset,' SET VAR2 1 MO/',adj,' = SUM2(t*BLK*',adj,')');
 writeln(dataset,' CLEAR SUM1 SUM2');
 writeln(dataset,'SPGRAPH(TWOSCALE)');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL ACTUAL vs. FORECAST
(Quarterly)''');
 writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Quarters'', $');
 writeln(dataset,'STYLE=',stile2,',VLABEL=''Forecast vs. Actual Levels'')');
 writeln(dataset,' # VAR1');
 writeln(dataset,'GRAPH 1');
 writeln(dataset,' # VAR2');
 writeln(dataset,'SPGRAPH(DONE)');
end;
writeln(dataset,' ');
if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
and ((dtype='A') or (dtype='a')) and (time_factor>=4) then
begin
 if(ttype='M') or (ttype='m') then adj:=12;
 if(ttype='Q') or (ttype='q') then adj:=3;
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NLA.GSP');
 writeln(dataset,'DO I=1,MO/',adj);
 writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
 writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
 writeln(dataset,'END DO I');
 writeln(dataset,' SET VAR1 1 MO/',adj,' = SUM1(t*BLK*',adj,')');
 writeln(dataset,' SET VAR2 1 MO/',adj,' = SUM2(t*BLK*',adj,')');
 writeln(dataset,' CLEAR SUM1 SUM2');
 writeln(dataset,'SPGRAPH(TWOSCALE)');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL ACTUAL vs. FORECAST
(Annual)''');
 writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Year'', $');
 writeln(dataset,'STYLE=',stile2,',VLABEL=''Forecast vs. Actual Levels'')');
 writeln(dataset,' # VAR1');
 writeln(dataset,'GRAPH 1');
```

```
 writeln(dataset,' # VAR2');
 writeln(dataset, SPGRAPH(DONE)');
end;
writeln(dataset,' ');
if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
and ((dtype='M') or (dtype='m')) then
begin
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NRM.GSP');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL RESIDUALS (Monthly)''');
 writeln(dataset,'GRAPH(HEADER=HEADER, $');
 writeln(dataset,'STYLE=',stile2,',VLABEL=''Difference Predicted vs. Actual'')');
 writeln(dataset,' # RESIDS');
end;
writeln(dataset,' ');

if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
and ((dtype='Q') or (dtype='q')) and (time_factor>=4) then
begin
 if(ttype='M') or (ttype='m') then adj:=3;
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NRQ.GSP');
 writeln(dataset,'DO I=1,MO/',adj);
 writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
 writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
 writeln(dataset,'END DO I');
 writeln(dataset,' SET VAR1 1 MO/',adj,' = SUM1(t*BLK*',adj,')-SUM2(t*BLK*',adj,')');
 writeln(dataset,' CLEAR SUM1 SUM2');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL RESIDUALS (Quarterly)''');
 writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Quarters'', $');
 writeln(dataset,'STYLE=',stile2,',VLABEL=''Difference Forecast vs. Actual Levels'')');
 writeln(dataset,' # VAR1');
end;
writeln(dataset,' ');

if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
and ((dtype='A') or (dtype='a')) and (time_factor>=4) then
begin
 if(ttype='M') or (ttype='m') then adj:=12;
 if(ttype='Q') or (ttype='q') then adj:=3;
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NRA.GSP');
 writeln(dataset,'DO I=1,MO/',adj);
```

```
  writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
  writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
  writeln(dataset,'END DO I');
  writeln(dataset,' SET VAR1 1 MO/',adj,' = SUM1(t*BLK*',adj,')-SUM2(t*BLK*',adj,')');
  writeln(dataset,' CLEAR SUM1 SUM2');
  writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL RESIDUALS (Annual)''');
  writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Year'', $');
  writeln(dataset,'STYLE=',stile2,',VLABEL=''Difference Forecast vs. Actual Levels'')');
  writeln(dataset,' # VAR1');
 end;
 writeln(dataset,' ');

end; {Procedure National}

procedure national_for;

 var adj    : integer;
stile1   : string[10];
stile2   : string[10];

begin
 stile1:='LINE';
 stile2:='LINE';
 if(style='b') or (style='B') then stile1:='BARGRAPH';
 if(style='b') or (style='B') then stile2:='OVERLAP';
 if(style='p') or (style='P') then stile1:='POLYGONAL';

 writeln(dataset,'IEVAL OUT = ',reg_end_yr,'-',reg_sta_yr);
 if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
 and ((dtype='M') or (dtype='m')) then
 begin
  right_trim(graph_path);
  if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NLM1.GSP');

  writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL GOALA vs. FORECAST ''');
  writeln(dataset,'GRAPH(HEADER=HEADER, $');
  writeln(dataset,'STYLE=',stile2,',VLABEL=''GOALA vs. Forecast'') 2');
  writeln(dataset,' # GOALA');
  writeln(dataset,' # CONTRACT');
 end;
 writeln(dataset,' ');

 if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
```

```
and ((dtype='Q') or (dtype='q')) and (time_factor>=4) then
begin
 adj:=1;
 if(ttype='M') or (ttype='m') then adj:=3;
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NLQ1.GSP');
 writeln(dataset,'DO I=((est_end/BLK)/',adj,')+1,MO/',adj);
 writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
 writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
 writeln(dataset,'END DO I');
 writeln(dataset,' SET VAR1 / = SUM1(t*BLK*',adj,')');
 writeln(dataset,' SET VAR2 / = SUM2(t*BLK*',adj,')');
 writeln(dataset,' CLEAR SUM1 SUM2');
 writeln(dataset,'SPGRAPH(TWOSCALE)');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL ACTUAL vs. FORECAST
(Quarterly)''');
 writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Quarters'', $');
 writeln(dataset,'STYLE=',stile2,',VLABEL=''Forecast vs. Actual Levels'')');
 writeln(dataset,' # VAR1');
 writeln(dataset,'GRAPH 1');
 writeln(dataset,' # VAR2');
 writeln(dataset,'SPGRAPH(DONE)');
end;
writeln(dataset,' ');
if((graph_for='Y') or (graph_for='y')) and ((nat='Y') or (nat='y'))
and ((dtype='A') or (dtype='a')) and (time_factor>=4) then
begin
 adj:=1;
 if(ttype='M') or (ttype='m') then adj:=12;
 it(ttype='Q') or (ttype='q') then adj:=3;
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NLA1.GSP');
 writeln(dataset,'DO I=((est_end/BLK)/',adj,')+1,MO/',adj);
 writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
 writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
 writeln(dataset,'END DO I');
 writeln(dataset,' SET VAR1 / = SUM1(t*BLK*',adj,')');
 writeln(dataset,' SET VAR2 / = SUM2(t*BLK* ,adj,')');
 writeln(dataset,' CLEAR SUM1 SUM2');
 writeln(dataset,'SPGRAPH(TWOSCALE)');
```

```
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL ACTUAL vs. FORECAST
(Annual)''');
 writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Year'', $');
 writeln(dataset,'STYLE=',stile2,',VLABEL=''Forecast vs. Actual Levels'')');
 writeln(dataset,' # VAR1');
 writeln(dataset,'GRAPH 1');
 writeln(dataset,' # VAR2');
 writeln(dataset,'SPGRAPH(DONE)');
end;
writeln(dataset,' ');
if((graph_resids='Y') or (graph_resids='y')) and ((nat='Y') or (nat='y'))
and ((dtype='M') or (dtype='m')) then
begin
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NRM1.GSP');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL RESIDUALS (Monthly)''');
 writeln(dataset,'GRAPH(HEADER=HEADER, $');
 writeln(dataset,'STYLE=',stile1,',VLABEL=''Difference Forecast vs. Actual'')');
 writeln(dataset,' # RESIDS');
end;
writeln(dataset,' ');

if((graph_resids='Y') or (graph_resids='y')) and ((nat='Y') or (nat='y'))
and ((dtype='Q') or (dtype='q')) and (time_factor>=4) then
begin
 adj:=1;
 if(ttype='M') or (ttype='m') then adj:=3;
 right_trim(graph_path);
 if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NRQ1.GSP');
 writeln(dataset,'DO I=((est_end/BLK)/',adj,')+1,MO/',adj);
 writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
 writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
 writeln(dataset,'END DO I');
 writeln(dataset,' SET VAR1 / = SUM1(t*BLK*',adj,')-SUM2(t*BLK*',adj,')');
 writeln(dataset,' CLEAR SUM1 SUM2');
 writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL RESIDUALS (Quarterly)''');
 writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Quarters'', $');
 writeln(dataset,'STYLE=',stile1,',VLABEL=''Difference Forecast vs. Actual Levels'')');
 writeln(dataset,' # VAR1');
end;
writeln(dataset,' ');

if((graph_resids='Y') or (graph_resids='y')) and ((nat='Y') or (nat='y'))
```

```
 and ((dtype='A') or (dtype='a')) and (time_factor>=4) then
 begin
  adj:=1;
  if(ttype='M') or (ttype='m') then adj:=12;
  if(ttype='Q') or (ttype='q') then adj:=3;
  right_trim(graph_path);
  if(length(graph_path)>1) then writeln(dataset,'OPEN PLOT ',graph_path,'\NRA1.GSP');
  writeln(dataset,'DO I=((est_end/BLK)/',adj,')+1,MO/',adj);
  writeln(dataset,' ACCUMULATE levelprj ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM1
((I-1)*BLK*',adj,')+1');
  writeln(dataset,' ACCUMULATE levelact ((I-1)*BLK*',adj,')+1 (I*BLK*',adj,') SUM2
((I-1)*BLK*',adj,')+1');
  writeln(dataset,'END DO I');
  writeln(dataset,' SET VAR1 / = SUM1(t*BLK*',adj,')-SUM2(t*BLK*',adj,')');
  writeln(dataset,' CLEAR SUM1 SUM2');
  writeln(dataset,' DISPLAY(STORE=HEADER) ''NATIONAL RESIDUALS (Annual)''');
  writeln(dataset,'GRAPH(HEADER=HEADER,HLABEL=''Year'', $');
  writeln(dataset,'STYLE=',stile1,',VLABEL=''Difference Forecast vs. Actual Levels'')');
  writeln(dataset,' # VAR1');
 end;
 writeln(dataset,' ');

end; {Procedure National Forecast}

procedure open_data;
label s1;
var
cont        : array[1..3] of string[2];
varlist_1   : string[70];
varlist_2   : string[70];
varlist_3   : string[70];
varlist_4   : string[70];
adj,ii      : integer;
v_flag1     : integer;
v_flag2     : integer;
v_flag3     : integer;
v_flag4     : integer;

begin

 right_trim(d_name);
 time_factor:=1;
 if(levels='y') then levels:='Y';
 if(deflator='y') then deflator:='Y';
 if(levels='n') then levels:='N';
```

```
if(deflator='n') then deflator:='N';
res3[1]:='RELPAY';
res3[2]:='UNEMP ';
res3[3]:='GOALA ';
res3[4]:='GOALNOTA';
res3[5]:='ARECPA';
res3[6]:='ADV';
res3[7]:='ACF';
res3[8]:='BON3';
res3[9]:='BON3SQ';
res3[10]:='BUYUP';
res3[11]:='POL86';
res3[12]:='DUM688';
res3[13]:='DUM489';
nvic_rat:=13;


if(ttype='M') or (ttype='m') then time_factor:=12;
if(ttype='Q') or (ttype='q') then time_factor:=4;
assign(dataset,'alloc.sys');
rewrite(dataset);
writeln(dataset,'BMA DATA 1500 GLOBAL 1000 COM 1000 CON 500 EXP 100 LOCAL 50
$');
writeln(dataset,'OPERANDS 100');
writeln(dataset,'OUTPUT NOECHO');
writeln(dataset,'CAL 78 1 12');
writeln(dataset,'IEVAL DBEGIN = (',reg_sta_yr,',',reg_sta_mo,')');
writeln(dataset,'IEVAL DLAST = (',reg_end_yr,',',reg_end_mo,')');
writeln(dataset,'IEVAL DLASTFC = (',for_end_yr,',',for_end_mo,')');
writeln(dataset,'IEVAL DFIRSTFC = DLAST + 1');
writeln(dataset,'IEVAL DEND = DLASTFC + 12');
writeln(dataset,'ALL 0 DEND');
writeln(dataset,'IEVAL MO  = DLAST-DBEGIN+1');
writeln(dataset,'IEVAL BLK = 1');
writeln(dataset,' ');
step:=0;

writeln(dataset,'OPEN DATA ',in_file);
v_flag1:=0;
v_flag2:=0;
v_flag3:=0;
v_flag4:=0;

cont[1]:='  ';
cont[2]:='  ';
```

```
cont[3]:=' ';
varlist_1:=' ';
varlist_2:=' ';
varlist_3:=' ';
varlist_4:=' ';
for k:=1 to nvic_rat do
begin
 if(copy(res3[k],1,1)<>' ') and (k<8) and (copy(inv[k],1,1)<>'C') then
 begin
  varlist_1:=varlist_1+res3[k]+' ';
  v_flag1:=1;
 end;
 if(copy(res3[k],1,1)<>' ') and (k>7) and (k<15) and (copy(inv[k],1,1)<>'C') then
 begin
  varlist_2:=varlist_2+res3[k]+' ';
  v_flag2:=1;
 end;
 if(copy(res3[k],1,1)<>' ') and (k>14) and (k<22) and (copy(inv[k],1,1)<>'C') then
 begin
  varlist_3:=varlist_3+res3[k]+' ';
  v_flag3:=1;
 end;

 if(copy(res3[k],1,1)<>' ') and (k>21) and (k<29) and (copy(inv[k],1,1)<>'C') then
 begin
  varlist_4:=varlist_4+res3[k]+' ';
  v_flag4:=1;
 end;
end;
if(v_flag2=1) then cont[1]:=' $';
if(v_flag3=1) then cont[2]:=' $';
if(v_flag4=1) then cont[3]:=' $';

writeln(dataset,' ');
 writeln(dataset,'DATA(FORMAT=RATS)  ',reg_sta_yr,',',reg_sta_mo,'
',reg_end_yr,',',reg_end_mo,' T126RM GSMARM');
 writeln(dataset,'SET(SCRATCH)  GSMARM  ',reg_sta_yr,',',reg_sta_mo,'
',reg_end_yr,',',reg_end_mo,' = LOG(T126RM(t)+GSMARM(t))');
 writeln(dataset,'DATA(FORMAT=RATS)  ',reg_sta_yr,',',reg_sta_mo,'
',reg_end_yr,',',reg_end_mo,' T126CM GSMACM');
 writeln(dataset,'SET(SCRATCH)  GSMACM  ',reg_sta_yr,',',reg_sta_mo,'
',reg_end_yr,',',reg_end_mo,' = LOG(T126CM(t)+GSMACM(t))');
 writeln(dataset,'DATA(FORMAT=RATS)  ',reg_sta_yr,',',reg_sta_mo,'
',reg_end_yr,',',reg_end_mo,' DAYSRM');
```

```
 writeln(dataset,'SET(SCRATCH) DAYSRM ',reg_sta_yr,',',reg_sta_mo,'
',for_end_yr,',',for_end_mo,' = DAYSRM(t-12)');
 writeln(dataset,'SET DAYS ',reg_sta_yr,' ',reg_sta_mo,' ',for_end_yr,',',for_end_mo,' =
LOG(DAYSRM(t))');

 writeln(dataset,' ');
 if(time_factor=12) then
 begin
  writeln(dataset,'IEVAL reg_sta = DBEGIN; IEVAL reg_end = DLASTFC');
 end;
 writeln(dataset,' ');
 writeln(dataset,'CLEAR PVWK1821 PVMILPAY RELPAY');
 writeln(dataset,'DATA(FORMAT=RATS) DBEGIN-2 DLASTFC+2 PVWK1821 PVMILPAY');
 writeln(dataset,'SET(SCRATCH) PVWK1821 DBEGIN DLASTFC = (PVWK1821(t-2) +
PVWK1821(t-1) + $');
 writeln(dataset,'                                PVWK1821(t)  + PVWK1821(t+1) + $');
 writeln(dataset,'                                PVWK1821(t+2))/5');
 writeln(dataset,'SET(SCRATCH) PVMILPAY DBEGIN DLASTFC = (PVMILPAY(t-2) +
PVMILPAY(t-1) + $');
 writeln(dataset,'                                PVMILPAY(t)  + PVMILPAY(t+1) + $');
 writeln(dataset,'                                PVMILPAY(t+2))/5');
 writeln(dataset,'SET RELPAY DBEGIN DLASTFC =
LOG(PVMILPAY(t)/(52*PVWK1821(t)))');
 writeln(dataset,' ');
 writeln(dataset,'CLEAR UNEMP');
 writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLASTFC UNEMP');
 writeln(dataset,'SET(SCRATCH) UNEMP DBEGIN DLASTFC = LOG(UNEMP(t))');
 writeln(dataset,' ');
 writeln(dataset,'CLEAR ACF');
 writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLAST ACFEXPT ACFCOV ACFPV');
 writeln(dataset,'SET(SCRATCH) ACFEXPT DFIRSTFC DLASTFC = ACFEXPT(t-12)');
 writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLASTFC COLLCOST');
 writeln(dataset,'SET ACF DBEGIN DLASTFC = LOG(ACFEXPT(t)/COLLCOST(t))');
 writeln(dataset,' ');
 writeln(dataset,'CLEAR BON3 BON3SQ BONUSQ BONUS3');
 writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLAST BONUS3 BONCOV3');
 writeln(dataset,'SET(SCRATCH) BONUS3 DFIRSTFC DLASTFC = (1.0975)*BONUS3(t-12)');
 writeln(dataset,'SET(SCRATCH) BONUS3 DBEGIN DLASTFC =
BONUS3(t)/(52*PVWK1821(t))');
 writeln(dataset,'SET(SCRATCH) BONCOV3 DFIRSTFC DLASTFC = BONCOV3(t-12)');
 writeln(dataset,'SET BON3 DBEGIN DLASTFC = BONUS3(t)*BONCOV3(t)');
 writeln(dataset,'SET BON3SQ DBEGIN DLASTFC = BON3(t)*BON3(t)');
 writeln(dataset,' ');
 writeln(dataset,'CLEAR GOALA GOALNOTA');
```

```
writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLASTFC AGLM13A AGLM3B
AROGOAL ARECPA');
writeln(dataset,'SET(SCRATCH) ARECPA DBEGIN DLASTFC = LOG(ARECPA(t))');
writeln(dataset,'SET GOALA    DBEGIN DLASTFC = LOG(AGLM13A(t))');
writeln(dataset,'SET GOALNOTA DBEGIN DLASTFC = LOG(AGLM3B(t)+AROGOAL(t))');
writeln(dataset,'CLEAR ADV');
writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLAST TV RADIO PRINT');
writeln(dataset,'SET(SCRATCH) TV    DFIRSTFC DLASTFC = (1.116)*TV(t-12)');
writeln(dataset,'SET(SCRATCH) RADIO DFIRSTFC DLASTFC = (1.116)*RADIO(t-12)');
writeln(dataset,'SET(SCRATCH) PRINT DFIRSTFC DLASTFC = (1.116)*PRINT(t-12)');
writeln(dataset,'DATA(FORMAT=RATS) DBEGIN DLASTFC ADVPDTV ADVPDRAD
ADVPDMAG');
writeln(dataset,'SET(SCRATCH) TV    DBEGIN DLASTFC = TV(t)/(ADVPDTV(t)*0.01)');
writeln(dataset,'SET(SCRATCH) RADIO DBEGIN DLASTFC =
RADIO(t)/(ADVPDRAD(t)*0.01)');
writeln(dataset,'SET(SCRATCH) PRINT DBEGIN DLASTFC =
PRINT(t)/(ADVPDMAG(t)*0.01)');
writeln(dataset,'SET ADV DBEGIN DLASTFC = LOG(TV(t)+RADIO(t)+PRINT(t))');
writeln(dataset,' ');
writeln(dataset,'CLEAR SEAS');
writeln(dataset,'SEASONAL SEAS 78.1 DLASTFC+12 12 78,3');
writeln(dataset,' ');
writeln(dataset,'CLEAR BUYUP');
writeln(dataset,'SET BUYUP DBEGIN DLASTFC = 0.0');
writeln(dataset,'SET BUYUP 87,6 DLAST = 1.0');
writeln(dataset,' ');
writeln(dataset,'CLEAR POL86');
writeln(dataset,'SET POL86 DBEGIN 85,9 = 0.0');
writeln(dataset,'SET POL86 85,10 DLASTFC = 1.0');
writeln(dataset,' ');
writeln(dataset,'CLEAR DUM489');
writeln(dataset,'SET DUM489 DBEGIN DLASTFC = 0.0');
writeln(dataset,'SET DUM489 89,4 89,4  = 1.0');
writeln(dataset,' ');
writeln(dataset,'CLEAR DUM688');
writeln(dataset,'SET DUM688 DBEGIN DLASTFC = 0.0');
writeln(dataset,'SET DUM688 88,6 88,6  = 1.0');
writeln(dataset,' ');
if((stats='Y') or (stats='y')) and
  ((sample='I') or (sample='i') or (sample='B') or (sample='b')) then
  begin
   writeln(dataset,'DOFOR I = $');
   writeln(dataset,'  RELPAY UNEMP GOALA GOALNOTA ARECPA ADV \CF BON3 $');
   writeln(dataset,'  BON3SQ BUYUP POL86 DUM688 DUM489');
   writeln(dataset,'STATISTICS I');
```

```
    writeln(dataset,'END DO');
    writeln(dataset,' ');
   end;
 raw:='y';
 if((trend='Y') or (trend='y')) and ((raw='Y') or (raw='y')) and
   ((sample='I') or (sample='i') or (sample='B') or (sample='b')) then stats_out;
 if((graph_run='Y') or (graph_run='y')) and ((raw='Y') or (raw='y')) then goto s1;
 writeln(dataset,' ');
 writeln(dataset,'EQUATION(MA=INPUT,MORE) 1 GSMACM');
 writeln(dataset,'# 1 2 3 7');
 writeln(dataset,'# RELPAY UNEMP GOALA GOALNOTA ARECPA ADV ACF BON3 3');
 writeln(dataset,'  BON3SQ BUYUP POL86 DUM688 DUM489 -SEAS -2 0 -SEAS -10 -7');
 writeln(dataset,'CLEAR RESIDS');
 writeln(dataset,'INITIAL(NOPRINT) 1');
 writeln(dataset,'ITERATE(ITERATIONS=50) 1 DBEGIN DLAST RESIDS');
 writeln(dataset,' ');
 writeln(dataset,'IEVAL N = DLASTFC - DFIRSTFC+1');
 writeln(dataset,'CLEAR CONTRACT');
 writeln(dataset,'FORECAST 1 N DFIRSTFC');
 writeln(dataset,'# 1 CONTRACT DFIRSTFC');
 writeln(dataset,'SET(SCRATCH) CONTRACT DFIRSTFC DLASTFC = EXP(CONTRACT(t)
+ 0.5 * RSS/NDF)');
 writeln(dataset,'ACCUMULATE CONTRACT DFIRSTFC DLASTFC CONSUM DFIRSTFC');
 writeln(dataset,'SET(SCRATCH) GOALA DFIRSTFC DLASTFC = EXP(GOALA(t))');
 writeln(dataset,'ACCUMULATE GOALA DFIRSTFC DLASTFC GOALSUM DFIRSTFC');
 writeln(dataset,'EVAL REDIN = GOALSUM(DLASTFC)/CONSUM(DLASTFC)');
 if(graph_for='Y') or (graph_for='y') then national_for;
 writeln(dataset,'WRITE ''Goal''');
 writeln(dataset,'WRITE GOALSUM(DLASTFC)');
 writeln(dataset,'WRITE ''Forecast''');
 writeln(dataset,'WRITE CONSUM(DLASTFC)');
 writeln(dataset,'WRITE ''REDIN''');
 writeln(dataset,'WRITE REDIN');

 writeln(dataset,' ');
s1: writeln(dataset,'END');
 close(dataset);
end; {open_data}

procedure recall_red;
begin
  assign(job5data,'red.sys');;
  {$i-}
  reset(job5data);
   {$i+}
```

```
 exist:= (ioresult = 0);
if exist then
begin
 reset(job5data);
 readln(job5data,tit);
 readln(job5data,method);
 readln(job5data,in_file);
 readln(job5data,model_file);
 readln(job5data,out_file);
 readln(job5data,rats_path);
 readln(job5data,graph_path);
 readln(job5data,nobs);
 readln(job5data,nent);
 readln(job5data,nper);
 readln(job5data,model);
 readln(job5data,ttype);
 readln(job5data,reg_sta_mo);
 readln(job5data,reg_sta_yr);
 readln(job5data,reg_end_mo);
 readln(job5data,reg_end_yr);
 readln(job5data,for_sta_mo);
 readln(job5data,for_sta_yr);
 readln(job5data,for_end_mo);
 readln(job5data,for_end_yr);
 readln(job5data,nvic);
 readln(job5data,nvic_rat);
 readln(job5data,trend);
 readln(job5data,stats);
 readln(job5data,graph_resids);
 readln(job5data,sample);
 readln(job5data,graph_for);
 readln(job5data,plot_for);
 readln(job5data,dtype);
 readln(job5data,nat);
 readln(job5data,dis);
 readln(job5data,graph_run);
 readln(job5data,levels);
 readln(job5data,deflator);
 readln(job5data,d_name);
 readln(job5data,style);
 readln(job5data,raw);
 for j:=1 to nvic do
 begin
  readln(job5data,res3[j]);
  readln(job5data,inv[j]);
```

```
   end;
   for j:=1 to 18 do
   begin
     readln(job5data,trans_in[j]);
   end;
   flag:=1;
   close(job5data);
  end;
 end;

begin
recall_red;
open_data;
end. {end of redfe}

writeln(xxx,'*************************************************************************
********');
writeln(xxx,'*  Modifications by Jack R. Dempsey Human Resources Research Organization *');
writeln(xxx,'*  Questions call (703) 549-3611                                          *');
writeln(xxx,'*************************************************************************
********');
writeln(xxx,'bma compile 1000 local 200  global 200 exp 320 ope 50 con 100');
writeln(xxx,'declare vector dates(10)');
writeln(xxx,'open data e_mdates.rcl');
writeln(xxx,'input(unit=data) dates');
writeln(xxx,'ieval dyr = fix(dates(1)) ; ieval dmonth = fix(dates(2))');
writeln(xxx,'       ieval freq = fix(dates(3))');
writeln(xxx,'calendar dyr dmonth freq');
writeln(xxx,'ieval dlastfc = cal(fix(dates(4)),fix(dates(5)))');
writeln(xxx,'write dlastfc');
writeln(xxx,'allocate 0 dlastfc+12');
writeln(xxx,'IEVAL DFROM = (70,1) ; IEVAL DTO = dlastfc-13 ;* dto= last data point');
writeln(xxx,'write dfrom dto                              ;* available for leading');
writeln(xxx,'ieval dto12 = dto+12                          ;* indicators.');
writeln(xxx,'write dto12');
writeln(xxx,'clear uermtos um1659');
writeln(xxx,'OPEN(BINARY) DATA ldindcat.rat');
writeln(xxx,'DATA(FORMAT=RATS) DFROM DTO lei1 lei5 lei7 lei8 lei19 lei20 lei21 lei27
lei28');
writeln(xxx,'data(format=rats) dfrom dto lei29 lei46 lei74 lei75 lei106 lei910 lei96');
writeln(xxx,'open data data_rrs.rat');
writeln(xxx,'clear allcivun');
writeln(xxx,'data(format=rats) dfrom dto+1 allcivun');
writeln(xxx,'set(SCRATCH) LEI5 dfrom dto = 10000.00/lei5(t)');
writeln(xxx,' ');
```

```
writeln(xxx,'IEVAL D1STONE  = DFROM - 12');
writeln(xxx,'IEVAL DLASTONE = DTO + 15');
writeln(xxx,'IEVAL DSEABASE = DFROM - 9');
writeln(xxx,'SEASONAL SEAS D1STONE Dlastfc+12 12 DSEABASE');
writeln(xxx,'set trend d1stone DLASTFC+12 = t');
writeln(xxx,' ');
writeln(xxx,'ieval d2 = dfrom + 1');
writeln(xxx,'ieval d3 = dfrom + 2');
writeln(xxx,'ieval start = d3 + 12');
writeln(xxx,' ');
writeln(xxx,'clear resids');
writeln(xxx,'statistics allcivun dfrom dto');
writeln(xxx,'LINREG allcivun dfrom dto resids');
writeln(xxx,'# constant trend');
writeln(xxx,'set ucycle dfrom dto = resids(t) + mean');
writeln(xxx,' ');
writeln(xxx,'clear resids');
writeln(xxx,' ');
writeln(xxx,' ');
writeln(xxx,'declare vector wt');
writeln(xxx,'dimension wt(17)');
writeln(xxx,'dofor i = LEI1 LEI5 LEI7 LEI8 LEI19 LEI20 LEI21 LEI27 LEI28 LEI29 LEI46
$');
writeln(xxx,'          LEI74 LEI75 LEI106 LEI910 LEI96');
writeln(xxx,'clear resids');
writeln(xxx,'LINREG(noprint) i dfrom dto resids');
writeln(xxx,'# constant trend -seas 0 -10');
writeln(xxx,'statistics i d2 dto');
writeln(xxx,' ');
writeln(xxx,'clear i');
writeln(xxx,'set i d2 dto = .7 * resids(t) + .3 * resids(t-1)');
writeln(xxx,' ');
writeln(xxx,'set i d2 dto = i(t) + mean');
writeln(xxx,'statistics(noprint) i d2 dto');
writeln(xxx,'eval sd = var ** 0.5');
writeln(xxx,'set i d2 dto = i(t)/sd');
writeln(xxx,' ');
writeln(xxx,'set(scratch) i d3 dto = (i(t) - i(t-1))/(.5 * (i(t) + i(t-1)))');
writeln(xxx,' ');
writeln(xxx,'end dofor');
writeln(xxx,' ');
writeln(xxx,' ');
writeln(xxx,'* Source E_ireg.rcl');
writeln(xxx,'* REVISED 9-6-85 (TO CORRECT SOURCE PROGRAMS)');
writeln(xxx,'* E_ireg.rcl to run regressions with individual leading series');
```

```
writeln(xxx,'*');
writeln(xxx,'*>>>>>>>forecasting <<<<<<<<<<<<<<<<<<');
writeln(xxx,'ieval locltime = dto');
writeln(xxx,'clear resids');
writeln(xxx,'equation(ar=input,ma=input,more) 1 allcivun');
writeln(xxx,'# 1');
writeln(xxx,'# 2 4');
writeln(xxx,'# -lei1 12 12 -lei5 12 12 -lei7 12 12 -lei8 12 12 -lei19 12 12 -lei20 12 12 $');
writeln(xxx,' -lei21 12 12 -lei27 12 12 -lei28 12 12 -lei46 12 12 -lei74 12 12 $');
writeln(xxx,' -lei75 12 12 -lei106 12 12 -lei910 12 12 -lei96 12 12');
writeln(xxx,'initial(noprint) 1');
writeln(xxx,'clear resids');
writeln(xxx,'iterate 1 72,5 locltime resids');
writeln(xxx,'* REVISED 9-6-85 (TO CHANGE FILENAME)');
writeln(xxx,'*Source P_autoco.rcl');
writeln(xxx,'*');
writeln(xxx,'clear autocor pautocor');
writeln(xxx,'correlate resids 0 0 24 autocor');
writeln(xxx,'partial resids 0 0 24 pautocor');
writeln(xxx,'plot(width=50,max=1,min=-1,number=1) 1');
writeln(xxx,'# autocor 1 25');
writeln(xxx,'plot(width=50,max=1,min=-1,number=1) 1');
writeln(xxx,'# pautocor 1 25');
writeln(xxx,'*');
writeln(xxx,'return');
writeln(xxx,'*SOURCE P_AUTOCO.RCL');
writeln(xxx,'*');
writeln(xxx,'clear fitted fcast errsq sumsq');
writeln(xxx,'forecast(print) 1 12 locltime+1');
writeln(xxx,'# 1 fcast locltime+1');
writeln(xxx,'errors(print) 1 12');
writeln(xxx,'# 1 sterrs locltime+1');
writeln(xxx,'set fcast locltime locltime = allcivun(t)');
writeln(xxx,'set(scratch) fcast locltime+1 locltime+12 = 0.7 * fcast(t) + 0.3 * fcast(t-1)');
writeln(xxx,'set(scratch) fcast locltime+13 locltime+13 = fcast(t-1)');
writeln(xxx,'set(scratch) fcast locltime+14 locltime+14 = fcast(t-2)');
writeln(xxx,'set(scratch) fcast locltime+15 locltime+15 = fcast(t-3)');
writeln(xxx,'set(scratch) fcast locltime+16 locltime+16 = fcast(t-4)');
writeln(xxx,'set(scratch) fcast locltime+17 locltime+17 = fcast(t-5)');
writeln(xxx,'set(scratch) fcast locltime+18 locltime+18 = fcast(t-6)');
writeln(xxx,'set upper90 locltime+1 locltime+12 = fcast(t) + 1.645 * sterrs(t)');
writeln(xxx,'set lower90 locltime+1 locltime+12 = fcast(t) - 1.645 * sterrs(t)');
writeln(xxx,'print(dates) locltime+1 locltime+12 lower90 fcast upper90');
writeln(xxx,' ');
writeln(xxx,'print(dates) dlastfc-11 dlastfc+5 fcast');
```

```
writeln(xxx,'clear newunemp');
writeln(xxx,'set newunemp dfrom dlastfc-12 = allcivun(t)');
writeln(xxx,'set newunemp dlastfc-11 dlastfc+5 = fcast(t)');
writeln(xxx,' ');
writeln(xxx,'dedit ldindcat.rat');
writeln(xxx,'delete ucycle');
writeln(xxx,'include ucycle ucycle 71,3 dto 1');
writeln(xxx,' cyclical component of unemployment');
writeln(xxx,' ');
writeln(xxx,'PRTDATA UCYCLE');
writeln(xxx,'SAVE');
writeln(xxx,'dedit data_rrs.rat');
writeln(xxx,'dreplace unemp newunemp 71,3 dlastfc+5');
writeln(xxx,'PRTDATA unemp');
writeln(xxx,'SAVE');
writeln(xxx,'end');
writeln(xxx,' ');
```


U. S. GOVERNMENT PRINTING OFFICE:1991-543-182/49012