

**COMPUTERIZATION OF REVENUE ALLOCATION INDICES (A CASE
STUDY OF NATIONAL REVENUE MOBILISATION, ALLOCATION &
FISCAL COMMISSION)**

By

**JOHN ANDA ⁹⁶
PGDMCS/304/97**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE, FEDERAL UNIVERSITY OF
TECHNOLOGY, MINNA. IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE AWARD OF A POST GRADUATE
DIPLOMA(PGD) COMPUTER SCIENCE.**

DECEMBER, 1999

TABLE OF CONTENTS

	Page
Certification -----	i
Inspiration -----	ii
Dedication -----	iii
Acknowledgement -----	iv
Abstract -----	v
CHAPTER ONE : INTRODUCTION	
1.1 Preamble -----	1
1.2 Powers and Functions -----	2
1.3 Operational Areas -----	3
1.4 Achievements -----	4
1.5 Aims and Objectives of the Project -----	5
1.6 Project Scope and Limitation -----	6
1.7 Data Collection -----	7
1.8 Types of Data to be Collected -----	7
1.9 Data Management -----	8-9
CHAPTER TWO : BACKGROUND INFORMATION	
2.1 Revenue Allocation -----	10
2.2 The Focus of Revenue Allocation -----	10-11
2.3 Principle of Revenue Allocation -----	12
2.4 National Revenue Mobilisation, Allocation & Fiscal Commission -----	13
2.5 Organogram of the National Revenue Mobilisation, Allocation & Fiscal Commission. -----	14 -16
2.6Evolving a stable Revenue Allocation Formulae and Principle -----	17-18
CHAPTER THREE : SYSTEM ANALYSIS	
3.1 Problem Identification and Definition -----	19
3.2 Feasibility Study -----	19
3.3 Testing Project Feasibility -----	19
3.4 System Analysis -----	20
3.5 Existing System -----	21
3.6 Problem with the Existing System -----	21
3.7 System Design -----	22
3.8 Cost and Benefits Analysis -----	23-25
CHAPTER FOUR : PROGRAM DEVELOPMENT/IMPLEMENTATION	
4.1 Language Choosing -----	26
4.2 Features of the Language -----	26
4.3 Input Specification -----	27-31
4.4 Output Specification -----	31
4.5 Program Design -----	31
4.6 Program Development -----	31
4.7 Program Description and Main Process -----	32-34
4.8 How to run the Program -----	34
4.9 System Requirements -----	35

4.10 Training needs	-----	36
4.11 System Conversion	-----	36-37
4.12 Review and Maintenance	-----	38

CHAPTER FIVE : SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary	-----	39
5.2 Conclusion	-----	39
5.3 Recommendation	-----	39-40
5.4 References	-----	41
5.5 Programs	-----	42-76
5.6 Output	-----	77-106

CERTIFICATION

This is to certify tht this project work was carried out by MR.JOHN ANDA in the department of Mathematics and Computer science in partial fulfilment of the post graduate dilpoma in Computer Science University of Technology, Minna.

.....
PRINCER R.BADAMOSI

.....
DATE

.....
DR.S.A.REJU
HEAD OF DEPARTMENT.

.....
DATE

.....
EXTERNAL EXAMINER

.....
DATE.

DEDICATION

This Project is dedicated to my son, **MASTER EMMANUEL ADINOYI ANDA**

ACKNOWLEDGEMENT

First and foremost, I ascribe the glory, honour and adoration to the Almighty God who has helped me hitherto.

I am inevitably indebted to my noble supervisor, PRINCE R. BADMOS whose supervision, hints,, facts and lectures have tremendously contributed to the success of my work.

I owe my regards to the Head of Department, Mathematics and Computer Science, ~~PROF. K. A. ADEBOYE~~ ^{DR S. A. ADEJU} whose lectures have also contributed immensely to the success of this project work. My gratitude also goes to the entire lecturers of the Department of mathematics and Computer Science whose guidance and various lectures contributed in no small measure to my writing this project.

I also wish to acknowledge the co-operation of the entire members of my immediate family, particularly my wife and my son for the love, tolerance and understanding during my absence from home when needed.

My gratitude also goes to my distinguished Director, in the person of KABIR ALHAJI BASHIR of National Revenue mobilisation, Allocation and Fiscal Commission, whose recommendations, word of advise and financial support led to the success of this programme.

Among my friends and colleagues who helped in one way or the other, special mention must be made of MR JOSEPH WUYEP, SUNDAY EKUKU, GBENGA DARAMOLA, MUHAMMED ONUKABA, RAHAMAN OTOYINOYI and ABUBAKAR UMAR of NNPC. I benefited immeasurably from their advise and support and I remain grateful to them.

I must not fail also to comment on the effort of MISS THEODORA ONYEBUCHI in giving this project work the best it deserves in terms of typing.

Conclusively, I must say that there are several other direct and less direct acknowledgement that should have been made in connection with the project but, due to the limitation imposed by time, space and for reasons beyond my control, I promise and pledge that such suggestions, and contributions will continue to live in my memory and most everlastingly in the mind of people who will be making use of this project work in the days to come. To people in this category, I give my highest regard.

My final gratitude goes to GOD, for I know that He loves and cares for me.

ABSTRACT

The project examines the revenue allocation principles which defined proxy measures for which relevant data are collected, computing the indices, storing and retrieval of the computed indices, by the National Revenue ,mobilisation , Allocation and Fiscal Commission .

The operations are done manually and problems associated with the system are that information can easily be revealed to unauthorised persons . The present manual collection, processing storing and retrieval of the computed indices are labour intensive , slow and costly . It is on this basis that the need for a computerised approach to this vital function of the commission .

The new system will be designed using Turbo Pascal 6.0 and Dbase iv programming language .This will enable the Commission to achieve good record management ,enhance speed , efficiency, storage facilities and retrieval of the information and the computed indices for quick submission to the Head of State for approval.

CHAPTER ONE:

INTRODUCTION

1.1.1 PREAMBLE

The issue of revenue allocation dates back to the Richards Constitution of 1946 which granted internal autonomy to the regions. The distribution of responsibilities between the Federal and Regional Governments was the beginning of revenue allocation effort in Nigeria.

Since 1946, Nigeria has had eight Revenue Commission and Committees and all were ad-hoc in nature and based on expediency. Each of these Commissions and Committees tried to address the principles behind revenue sharing among the components part of Nigeria however, whatever recommendation the Commissions and the Committees came up with, there has been a new wave of agitation demanding different methods of sharing or allocation.

The following Commissions and Committees with interjecting decrees to existing sharing or allocation methods were set up since 1946:

_Phillipson Commission	(1946)
_Hicks-Phillipson Commission	(1951)
_Click Commission	(1953)
_Raisman Commission	(1958)
_Binns Commission	(1964)
_Decree No. 15	(1967)
_Dina Committee	(1969)
_Decree No.13	(1970)
_Decree No.9	(1971)
_Decree No. 6	(1975)
_Aboyade Committee	(1977)

_ Okigbo Commission	(1980)
_ Revenue Allocation Act	(1981)
_ Revenue Allocation Amendment Decree	(1989)
_ Decree No. 49	(1989)

It was the Aboyade Committee of 1977 that a permanent body was first brought into focus . Since then, others had made an impact on the politics of revenue allocation . By early 1989, the Economic Affairs Department of the Presidency (ECA) collated all past records and recommendations including their white papers regarding the setting-up of Fiscal Commission and a memorandum was submitted to the Federal Government. The Armed Forces Ruling Council (AFRC) approved the memorandum in 1989, and the details of the composition, functions etc. were sent to the Federal Ministry of Justice for legal authority, hence the birth of Decree no. 49 of 1989 and the coming into existence of National Revenue Mobilisation, Allocation and Fiscal Commission.

1.1.2 **POWERS AND FUNCTIONS**

The Commission is charged with the following functions and responsibilities :

- (a) Monitor the accruals to and disbursement of revenue from the Federation Account.
- (b) Review, from time to time , the revenue allocation formulae and principles in operation to ensure conformity with changing realities .
- (c) Advise the Federal, States and local Governments on fiscal efficiency and methods by which their revenue is to be increased.
- (d) Make recommendations and submit its finding by a report to the Government of the Federal or to the States, as the case may be, regarding the formula for the distribution of the Federal accounts and the local governments accounts.

- (e) Collaboration with all layers of governments as well as their ministries, departments, agencies and extra-ministerial units in the prompts, and faithful production of public financial statistics.
- (f) Determination of the remuneration which it may deem appropriate for political office holders, such members of the executive and legislative branches of government outside the consolidated account.
- (g) Commissioning, undertaking or sponsoring studies, analysis and deliberations on subjects which may bear directly or infringe significantly on the policy and operation domain of Federal fiscal system and inter-governmental financial relations.
- (h) Making whatsoever general or specific recommendations as the Commission may consider necessary for more effective mobilisation, collection, allocation and distribution of federal, states and local government revenues, as well as providing guidelines for their efficient implementation; and
- (i) Submitting regular and timely annual reports to the federal government on its general activities over and beyond its specific recommendations or ad-hoc submission on particular subjects, with such annual reports also incorporating the Commission's audited accounts.

1.1.3 **OPERATIONAL AREAS**

The Commission has five functional departments and the internal audit unit. These are :

- (a) Allocation Department
- (b) Mobilisation Department
- (c) Planning, Research and Statistics Department
- (d) Personnel Management Department
- (e) Finance and supply Department

Also the Commission has seven zonal offices that takes charge of field operations, and enhance co-ordination and contacts with states and local governments. These seven zonal offices and the states they cover are as follows :

- (a) LAGOS : Lagos, Ogun, Oyo, Kwara
- (b) KANO: Kano, Jigawa, Katsina, Sokoto, Kebbi and Zamfara
- (c) PORT-HARCOURT: Rivers, Bayelsa, Cross-River and Akwa-Ibom
- (d) YOLA: Adamawa, Taraba, Borno and Yobe
- (e) ENUGU: Enugu, Anambra, Imo, Abia, Ebonyi, Benue and Kogi
- (f) JOS: Plateau, Nassarawa, Bauchi, Gombe, Kaduna and Niger
- (g) AKURE: Ondo, Ekiti, Edo, Delta and Osun.

1.4 ACHIEVEMENTS

The Commission, since its establishment in 1989 has made giant strides in all the areas prescribed by the laws and status establishing it. The first revenue allocation formula prepared by the Commission was approved with minor changes by the Armed Forces Ruling Council (AFRC) and it became operational with effect from January 1990. The current revenue allocation formula in use, is slightly different from the one which came into effect in 1990 due to review undertaken between 1992 and 1993 by the Commission.

Representatives of the Commission attends monthly Federation Accounts Allocation Committee (FAAC) meetings as an observer to ensure that allocation of funds from the federation account is done as per the approved revenue allocation formula. The Commission is also a member of the Board of the Federal inland Revenue Services, the National Council on Statistics, and has observer status on the Joint Tax Board.

The Commission, in 1997, undertook a nation-wide verification exercise of some data used in revenue allocation to the states and local governments and submitted its

report to the government for consideration and approval and adoption in the allocation of revenue to states and local governments in 1998.

The involvement of the Commission in the writing of the white paper on the report of the Committee on Modified Value Added Tax (MVAT) as a form of taxation and subsequent introduction of VAT, a tax that has considerably increased the revenues of federal,, states and local governments is another land mark in the history of the Commission.

The Commission has also been concerned with designing appropriate strategies and policies in the area of revenue mobilisation and fiscal policy. In the year 1998, the Commission was given approval to start its monitoring duties and is presently monitoring some federal revenue generating agencies like NNPC and custom services.

Through its various market studies , the Commission has been advising local governments on ways to improve their revenue from some of these markets. Consequently, this has created awareness among the local governments that substantial revenues can accrue from markets.

On the remuneration of political office holders, this function was transferred to the Commission to avoid the recurrence of the sad experience in the second Republic when legislators fixed their salaries and allowances in manner that was considered outrageous. As a technical body, the Commission produced and submitted a report to the government in 1989 and it was the recommendations contained in the report that were used to arrive at present salary structure of political office holders. In 1998, a review was undertaken by the Commission and a report was submitted to the government for consideration, approval and adoption in the fourth Republic.

1.5 **AIMS AND OBJECTIVES OF THIS PROJECT**

Wrong policy making may result from lack of necessary data on which to base decision. Good planning can only be achieved with relevant and accurate data. One of the major problems that affect timeliness in data delivery is low capacity for processing and analysis of data due to lack of resources and expertise. This twin issues need to be tackled

if the end product is to be meaningful. The aims and objectives of this project therefore are :-

- (i) To study in details, the method of data collection, processing, storing and retrieval of the processed data/indices by the National Revenue Mobilisation, Allocation and Fiscal Commission.
- (ii) Design means and strategies required to effect computerisation of the revenue allocation indices on theoretical basis so that in the near future, this can be used for the same project.
- (iii) Design and implementation of the soft ware required for effective use in the near future.
- (iv) To exploit the available facilities in a digital computer system in the computation of revenue allocation indices.

1.6 **PROJECT SCOPE AND LIMITATIONS**

The project covers the basic principles which are used in revenue allocation indices computation. The new system is near perfection in its operations and will work efficiently in the specified environment in Chapter three (System Analysis). The computer based revenue allocation indices to be introduced would automate the indices used for revenue allocation. The limitation in scope does not necessarily imply that certain things cannot be generalised in terms of application and reference .

Due to time constraints and in-availability of source information, the soft ware developed did not take care of the principle used in allocating the 3% of oil revenue due to oil producing states.

1.7 DATA COLLECTION

Collection of data constitutes the first step in any statistical investigation, utmost care must be exercised in collecting data because they constitute the foundation of analysis. If they are faulty, the conclusion drawn can hardly be dependable. Data may be expressly collected for a specific purpose, such data are known as primary data . The direct collection of facts and figures relating to the primary and secondary school enrolment provides primary data. The great advantage of such data is that the exact information wanted is obtained. Terms are carefully defined so that , as far as humanly possible, misunderstanding is avoided.

Often, however, data collected for some other purposes may be used. Such data is known as secondary data. Details of mean annual rainfall by states are compiled by the meteorological department of Federal Ministry of Aviation. This is used, however, for computing the indices on rainfall proportion as one of the social development factors. Secondary data must be used with great care, this is because such data may not give the exact kind of information wanted and may not be in the most suitable form. Great attention must be paid to the precise coverage of all information in the form of secondary data. Before such data can be used, it will be necessary to know the source, how they are obtained, and method of compilation.

1.8 TYPES OF DATA TO BE COLLECTED

Each revenue allocation principle has defined proxy measures for which the relevant data has to be obtained. Data are required in the following:

- _ Number of local governments in a state to be used in allocating revenue to local government in a state under the principle of equality of local government.
- _ the population figures (based on 1991 census figures) of each state or local government

_internally generated revenue of each state and local government in the country. This is based on the incremental concept which is defined as the percentage increase in internally generated revenue in a given period over the preceding period.

_For the social development factor principle, data is required in the following:-

- (a) Primary school enrolment figures
- (b) Secondary/ Commercial school enrolment figures
- (c) Number of hospital beds or health clinic
- (d) mean annual rainfall of each state capital or local government headquarters
- (e) Territorial spread of each state or local government.

_ The land mass of each state and local government.

_ The terrain type of each state and local government.

1.9 DATA MANAGEMENT

The introduction of computers in organisations and the ever increasing sophistication of data processing system have highlighted the importance of data as one of the most valuable organisational resources. It is from the manipulation and interpretation of data that information is generated and in turn, used in the decision making process . The realisation of this importance of data calls for proper management and efficient organisation of the data.

Management information system is an organised collection of people, procedures, data base and devices used to provide routine information to managers and decision makers. The focus of management information system is an operational efficiency.

Managerial decision making can be classified into three categories; strategic, tactical and operating decision making. Management information system aims to help managers make effective decisions by providing them with reports or information that are

timely and to the point. The management information system should aim and fulfil the followings:-

- (i) Relevance: Information provided should be relevant to the decision maker.
- (ii) Accuracy: Data that enters into the system must be validated to ensure that decision are made with information obtained from accurate data.
- (iii) Timeliness: The information must reach the man at the time it is most needed and useful.

CHAPTER TWO:

BACKGROUND INFORMATION

2.1 REVENUE ALLOCATION

Revenue allocation refers to the mechanism for sharing the country's financial resources among the different tiers of government in the federation, with the overall objective of enhancing economic growth and development, minimising inter-governmental tensions and promoting national unity. In Nigeria, this involves the sharing of national revenue and other resources first vertically and secondly, horizontally among the states and among the local governments.

The distribution of federation finance is not restricted to direct allocation to governments. Direct allocation can be made for identified project/programmes to be managed invariably by independent agencies. This type of allocation is often referred to as "Special Fund". The funds could be used for development of special areas or to deal with special problems which are regarded as national importance. The presumption being that such problems and /or area might not be effectively handled if left to a particular government. Generally, a satisfactory revenue allocation system should strive to satisfy the twin objectives of equity and efficiency, strengthen the effort of the federal government to manage the national economy, ensure adequate financial flows to the states and local governments and be capable of quickly adapting to changing circumstances.

2.2 THE FOCUS OF REVENUE ALLOCATION ARRANGEMENT

- (1) Federalism set up different levels of government and assigns constitutional responsibilities to these different levels. The constitutional responsibilities

create different expenditure obligations for these different levels of government. In a federalism, therefore, along with the assignment of responsibilities, the issues connected with financial arrangements have to be sorted out.

- (2) The logic underlying the allocation of tax powers does not tally with the logic underlying the assignment of constitutional responsibilities; there exists an imbalance between the expenditure obligations and the revenue sources of these levels in every federation. A higher level of government may have revenue advantage over lower-level of governments, as a result of the powers conferred on it over certain tax revenues. Revenue allocation therefore, has been evolved as a mechanism for redressing this imbalance between expenditure obligations and revenue sources.
- (3) Revenue allocation is also focused at increasing the level of provision of activities that give rise to external benefits of ensuring minimum level of provision of certain activities. Such allocations are commonly referred to as "Grants-in-aid".
- (4) The National political units are unequal and their capacities are unequal, an allocation based on strict equality will exacerbate existing inequalities hence the equity objective, as opposed to equality, to achieve socio-economic justice among states/local governments. If the principle of equality is adopted by treating each unit equally irrespective of the disparity in their endowments, the result will be the transmission of inequalities from one generation to another.

2.3 PRINCIPLES OF REVENUE ALLOCATION

Principle of revenue allocation are the rules or factors to be applied in revenue allocation. Basically, there are five main principles used in revenue allocation among the states/local government . These principles are:

- (1) **EQUALITY OF STATE/LOCAL GOVERNMENT** : This principle is designed to take care of the minimum responsibilities that each constituent state/local government has to constitutionally carry out regardless of its size.
- (2) **POPULATION** : The population principle is designed to take care of many dimensions of human needs.
- (3) **SOCIAL DEVELOPMENT FACTOR** : The social development factor is designed as the totality of changes that contribute positively to the mental, physical and psychological welfare of a nation's citizenry. Indicators of social development factor includes education, health delivery and water supply.
- (4) **LAND MASS AND TERRAIN** : The land mass factor was introduced to take care of developmental and administrative costs and problem associated with large land and areas where available resources have to spread out over a large expanse of land. On the other hand, the terrain factor has been introduced to take care of huge developmental costs associated with physical constraints, such as; steep slopes, swampy etc.
- (5) **INTERNAL REVENUE EFFORT**: The proxy measure used for the internal revenue effort is incremental concept. This is the percentage increase in internally generated revenue in a given revenue period over the preceding.

Generally, the above principles represent a combination of equity and efficiency measures, allowing for the minimisation of the development gaps among states/local governments, the absorptive capacity of states/local governments to make full use of funds, the ability of each tier of government to realise its potential taxable capacity in respect of these revenue sources within its jurisdiction and finally, the minimisation of the costs of operating government functions.

2.4 NATIONAL REVENUE MOBILISATION, ALLOCATION AND FISCAL COMMISSION.

Prior to the establishment in 1989 of the National Revenue Mobilisation Allocation and Fiscal Commission as a permanent body to undertake on a continuous basis, the review of the fiscal relationship between the various levels of government, Nigeria has had eight Revenue Commission and Committees and all were ad-hoc in nature. The major achievements of these early efforts at revenue allocation include:

- (a) The establishment of regional/state government autonomy over certain revenue sources (e.g. personal income tax).
- (b) The establishment of federal government exclusive control of some revenue sources (e.g. armed forces income tax).
- (c) The creation of a distributive pool account into which other revenues (including import and export taxes, mining rents and royalties etc) were paid and which was subsequently distributed between the federal and regional/state governments.

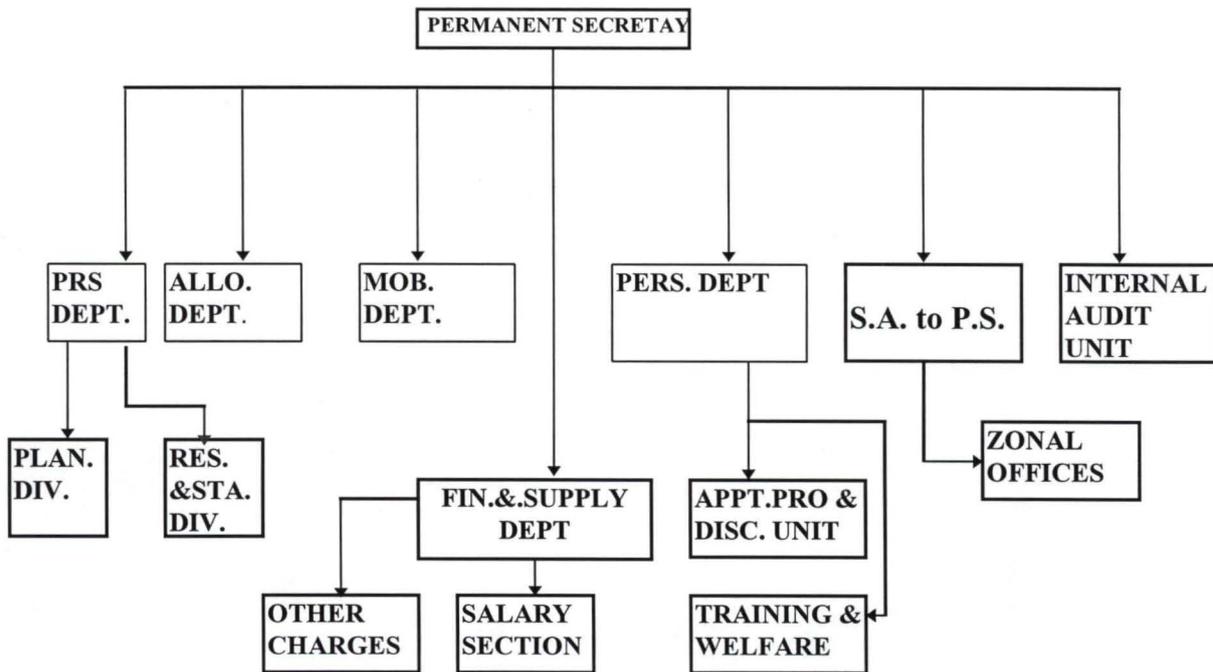
It was the Aboyade Committee of 1977 that a permanent body was first brought into focus. Since then, others who had made an impact on the politics of revenue allocation includes the

Commission of 1980, the C.S Cookery Bureau of 1987, the Economic Affairs (ECA)

department of the Secretary to the Government of the Federation's Office that prepared Council Memorandum in 1989 which was approved by the AFRC, thus setting up the National Revenue Mobilisation, Allocation and Fiscal Commission.

**2.5 ORGANOGRAM OF NATIONAL REVENUE
MOBILISATION, ALLOCATION AND FISCAL COMMISSION**

The current Organogram of National Revenue Mobilisation, Allocation and Fiscal Commission is shown below:



2.5 FUNCTIONS AND RESPONSIBILITIES OF EACH DEPARTMENT

a) **PLANNING, RESEARCH AND STATISTICS DEPT.**

- i.) Formulating plans and programmes for the implementation of the commission's policies;
- ii.) Designing and co-ordinating field researches for the Commission .
- iii.) Analysis and documentation of research data for the Commission's guide.
- iv.) Publication and issuance of periodic statistical information and reports .
- v.) Liaison with appropriate research institutions, universities, financial institutions and government agencies and other international bodies for the Commission.
- vi.) Preparation of the capital budget and rolling plan as well as monitoring and evaluation of plan implementation.
- vii) Secretariat of the Management Tenders Board.

b) **ALLOCATION DEPARTMENT**

- i) Co-ordination of the processes leading to the prescription and application of the revenue allocation formula for the sharing of the federation account between the federal, states and local governments.
- ii) Co-ordination of the periodic revenue allocation formula.
- iii) Co-ordinating the collection, collation and analysis of data for the computation of indices used in revenue allocation for the federation account.
- iv) Co-ordination and monitoring of the disbursements from the federation account and the various accounts and special accounts that may be established by the Federal Government .

- v) Attending the meetings of the federation accounts allocation Committee.

(c) FINANCE AND SUPPLY DEPARTMENT

- i) The department, in line with the civil service regulation is the custodian of funds allocated to the Commission by the Budget Office. In addition to keeping the money, it also disburses the fund to other departments based on the approval of the accounting officer.

(d) PERSONNL DEPARTMENT:

_ APPOINTMENT, PROMOTION DISCIPLINE:-

- i) Processing employment/appointments and promotions of the staff of the Commission.
- ii) Takes charge of the Registry and personal records of the Commission.
- iii) Prepares the Commission's annual manpower budget.
- iv) Posting and deployment of staff within the Commission.
- v) Serving as Secretariat of the two personnel management committees.

_ TRAINING AND WELFARE

- i) Identification of training needs of the Commission.
- ii) Preparation of annual training proposals and budget of the Commission.
- iii) Defence of the Commission's training proposals before the office of the Establishments and Management Services.
- iv) Implementation of approved training programme.

- v) All matters relating to identification of, and provision of residential accommodation to entitled members of staff.
- vi) General maintenance of the official residence of staff.
- vii) Processing of request for all advances and loans.

2.6 EVOLVING A STABLE REVENUE ALLOCATION FORMULAE AND PRINCIPLE

The issues here is whether it is possible to adjust, possibly reduce the revenue items in the account and assign such to states and or local governments. It is important to first briefly consider the federation account which is essentially what is being allocated to the various tiers of government. Federation account is made up of revenue from the following sources :

- (a) Company income tax
- (b) Import duties
- (c) Export duties
- (d) Excise duties
- (e) Petroleum profits tax
- (f) Mining rents and royalties
- (g) NNPC earnings from direct sales
- (h) Pipeline licences and other fees; and
- (i) Surplus arising from the sales of gas.

The Federation account is provided for in section 160 of the 1989 Constitution. According to this section:

- (i) “ The Federation shall maintain an account to be called ‘the federation account’ into which shall be paid all revenues collected by the government of the Federation, except the proceeds of the personal income tax from the residents of the F.C.T. Abuja”
- (ii) “Any amount standing to the credit of the federation account shall be distributed among the federal, state and the local governments, in each state, on such terms and in such manner as may be prescribed by the National Assembly.”
- (iii) “Any amount standing to the credit of the states in the federation account shall be distributed among the states on such terms and in such a manner as may be prescribed by the national Assembly.”
- (iv) “Any amount standing to the credit of the local government in the federation account shall be allocated directly to the local governments concerned on such terms and in such a manner as may be prescribed by the National Assembly.”

It should be noted however that re-assignment of responsibilities to match revenues may involve assigning to a lower level of government responsibilities that would, otherwise, be more suitable for higher levels of government.

Also, a re-alignment of revenue powers to match responsibilities may involve transferring to lower levels of government revenue sources that cannot efficiently be administered. From this, one can say that the relative stability of the allocating of responsibilities and the assignment of revenue jurisdiction under Nigerian fiscal system may have been informed by the fact that not much headway could be made in the direction of re-assignment of responsibilities and revenue powers.

CHAPTER THREE:

SYSTEM ANALYSIS

3.1 PROBLEM IDENTIFICATION AND DEFINITION

Revenue allocation indices used for the distribution of federation account between the states and local governments are computed manually by the National Revenue Mobilisation, Allocation and Fiscal Commission. The problem associated with this manual method of computations are that information can easily be revealed to the unauthorised person, files containing the data/information about each state and local government are kept in offices cabinets and shelves and retrieval of such files are some times difficult and time consuming. The manual computation of the indices are also laborious.

The problem now is to seek and develop a system that would ease computation, storage and retrieval of the indices, thereby eliminating or at least drastically reducing all the draw-backs associated with the manual method.

Computerisation of revenue allocation indices has been identified to be the only solution to the above problems. By computerisation, the indices will be automated and the review of the indices as the need arise can be carried out with ease.

3.2 FEASIBILITY STUDY

This stage which has been embarked upon is to carefully examine the current manual way of computing the revenue allocation indices, method of storing the computed indices and its retrieval with a view to determining whether or not the proposed system is desirable.

3.3 TESTING PROJECT FEASIBILITY

The project feasibility has to be tested. This is to prevent wasting of many months of efforts and many thousand of naira if the project is too large, too uncontrollable or

simply impossible to carry out. The testing of feasibility study is a miniature system analysis and design effort that entails an exploration of alternatives seen to be realistic in their potential costs and benefits, the project proceeds to the next phase, system analysis. Project feasibility can be tested in many ways:-

- (A) OPERATIONAL FEASIBILITY :We have to look into the workability of the proposed system design, when it is developed and subsequently installed.
- (B) TECHNICAL FEASIBILITY: Here, it is necessary to find out or clarify if the system designed can be ran with the right equipment in terms of the soft ware, personal or even the hardware .
- (C) FINANCIAL OR ECONOMIC FEASIBILITY: This is to check the cost of implementing the new system to be designed and also the benefits derived from the new system.

In carrying out an analysis of a system, there are some steps to be taken. Firstly, an investigation of the current system needs to be made. The objectives of the organisation must be considered and the strengths and weaknesses must be brought to light, on which the development of new system could be based.

3.4 **SYSTEM ANALYSIS**

System analysis can be defined simply as the organisation of data/information gathered about the existing system into a meaningful form. In other words, system analysis can be defined as the study of an existing system in order to provide detail information about the nature, characteristics, problems and weakness of the system which will serve as a basis for designing a more reliable system.

System analysis is an important intermediate stage between system investigation and system design for the proposed system. At the end of any system analysis, there should be a report known as requirement specification. The specification of requirement

simply spell out every possible solutions to overcome the problems among the users of the existing system.

Here certain vital questions are asked;

- (a) Why did the problem occur ?
- (b) Why were the present methods adopted ?
- (c) What are the alternative method ?

3.5 **EXISTING SYSTEM**

The revenue allocation indices are computed manually by the staff of the National Revenue Mobilisation, Allocation and Fiscal Commission . The only means of storing the data collected is to open a file for each of the states and local government and these files are kept in file cabinets.

To retrieve any record, the office clerk has to search for the appropriate file in the file cabinets. In addition to the above, the following actions are also performed manually. They are:-

- (i) Data collection
- (ii) Data verification
- (iii) Data collation
- (iv) Tabulation

3.6 **PROBLEM WITH THE EXISTING SYSTEM**

The existing system has the following problems:

- (a) Storage Problem : Information stored in files make the file appear bulky and occupy more space in the cabinet, at times, no enough storage facility.

- (b) Retrieval Problems : The speed of retrieving the information is very slow as one has to search for a particular file (among many) from which the information is to be obtained.
- (c) Poor safety and security for the files.
- (d) Poor file handling resulting in destruction of vital documents.
- (e) Reported cases of missing files.
- (f) Ineffective management information system.

In order to appreciate the role of the Commission by the country's populace and the lower two tiers of governments in particular, for computing the revenue allocation indices, this system needs to be improved to the required standard. To remove the problems highlighted above, an advanced technology like computer system is required.

3.7 SYSTEM DESIGN

The specification requirements is an input to the system design. The purpose of the design stage is to work from the requirements specification to produce a system specification. The system specification will be a detailed set of input documents, forms, report layout which provide details of all features of the system.

The system design is divided into two phases , the logical design and the physical design. The logical design transform the specification of both data and processing requirement to obtain a detailed requirements of the proposed system. The physical design transforms the logical specification database and program specification. During the physical design, details such as input and output formats will be designed along with file format, screen format and so on .

The objectives of the system design includes efficiency and flexibility . Efficiency involves accuracy and timeliness of the system out put . Flexibility means that the

designed system should be responsive to the changes inevitably requested by its user in the nearest future.

3.8 COST AND BENEFITS ANALYSIS

_ OPERATING COST

=N=

Supplies for one year	100,000	=
Equipment maintenance	50,000	=
Installation	30,000	=
Program maintenance	10,000	=
Labour cost (five operators)	15,000	=
Fire extinguisher	15,000	=
Miscellaneous	20,000	=

Total =N=240,000

_ DEVELOPMENT COST

=N=

System analysis and design	50,000	=
Software development-Implementation	20,000	=
Equipment purchases egg. 2 P. C.	250,000	=

UPS (3500 VHS)	15,000	=
Printers (2) Laser Jet 5 L	100,000	=
Personnel training	50,000	=
Miscellaneous	25,000	=
Grand Total =N=	<u>=N=510,000</u>	

BENEFITS

The benefits accruing from the proposed system includes :

- (1) **SPEED** : The most obvious benefits of using a computer is speed. The computer can perform calculations and data processing more quickly than the manual method. Work that might take one several months to complete manually can be accomplished in hours or at most days by computer . The proposed system will therefore leads to quick submission of the computed indices to the Head of State for approval and subsequent implementation .
- (2) **ELIMINATION OF HUMAN ERROR**: Manual calculations are often subject to human errors. The proposed system will eliminate all these errors and move up to the error free computation.
- (3) **IMPROVED SYSTEM SECURITY** :With the proposed system, each authorised user is assigned a unique password which is encrypted and stored away in hidden file which are in-accessible, even to the system administrator . As a result, the new system presents a very secure working environment and ensures that unauthorised person never have access to this important and security documents.

- (4) LESS PAPER AND PAPER WORK: One major highlight of computerisation is the fact that information is stored in magnetic disks as against paper. One magnetic disk can store information equivalent to millions of pages of paper. As such, a lot of space hitherto taken up by large filing cabinets can now be put to better use.
- (5) RETENTION: The computer can store and search massive files of data and programs. The contents of the files does not fade ,and it can be used as often as needed .
- (6) ECONOMY : The advantages of speed and accuracy can often be translated into naira savings . Usually the per unit cost of processing data by computers is considerably lower than by manual means.
- (7) FLEXIBILITY: The new system is designed such that it can operate in a dynamic terrain rather than static environment .
- (8) REDUCE MANPOWER REQUIREMENTS : Since most of the calculation work will now be done by the computer , the office will need less human effort to 'forge' ahead. Staffs who have been engaged in the manual computation of the revenue allocation indices annually can now be given a more fruitful and exciting schedule. This will enable the Commission bring a new lease of life to the discharge of its responsibilities.

CHAPTER FOUR

PROGRAM DEVELOPMENT/IMPLEMENTATION

4.1 LANGUAGE CHOSEN : The language chosen for this project work is dbase IV . Database Management System (DBMS) is a soft ware that constructs, expands and maintains data contained in a database. It also provides the interface between the user and the data in such a way that it enables the user to record, organise, select, summarise, extract, report on, and otherwise, manage the data contained in a database. Database management system programs keep information in files, and within each file in a collection of related information.

4.2 FEATURES OF THE LANGUAGE (DBASE IV)

The features and objectives of database which reflect the present needs of the modern data processing community are as follows:-

- (a) **Data Integration is achieved:-** In a database, information from several files is coordinated, accessed and operated upon as though it is in a single file. Logically, the information is centralised, physically, the data may be located on different devices. In addition, it is possible for two or more application to be sharing compatible data. This allows the users to gain valuable information by linking data across the organisation. Data integration is achieved by using the technique such as indexing.

- (b) **Data Redundancy is Reduced or Eliminated:-** Data redundancy occurs in file processing systems when the data cannot be arranged to suit all the application program accessing these data. This results in the same data appearing in more than one file depending on the application program that request the use of the data. The duplication of data leads to wastage of storage space and duplication of efforts during data entry. This is advantageous in a database environment because, similar data are not duplicated, they are stored once.

(c) **Data Independence Can be achieved:-** Applications evolve as more information is required, and its usage improves. If any changes occur to the data records during the life of the file, then all programs accessing these data must be changed. However, database system provides data independence of program. Data independence is the insulation of application programs from the changing aspects of physical data organisation. This feature seeks to allow changes in the content and organisation of physical data without re-programming of applications, as well as allowing for modifications to application programs without re-organising the physical data.

(d) **Data Integrity Can Be Achieved:-** In an environment where there is duplication of data, problems concerning the updating and deletion of data will necessitate a change in every file that contains the same data, otherwise, inconsistency will result. In a database system, this problem can be avoided by recording data once. This approach to data integrity results in more consistent information. One update being sufficient to achieve a new record status for all applications which use the data.

(e) **Data Are Centrally Controlled :-** in database environments, data and operations on data are centrally controlled and this can lead to a better management of data by enforcing standards for all the database users on how information would be released out. Specifically, it allows for proper security of data since there is only one source of data in the organisation and such standards would easily be enforced than an environment where data are duplicated.

4.3 INPUT SPECIFICATION

There are two sections for data entry, namely: state government and local government. The entry is via the keyboard. Each of these sections has eleven categories, namely:

(i) population

(ii) Land mass

- (iii) Secondary school enrolment (D)
- (iv) Secondary school enrolment (I)
- (v) Primary school enrolment (D)
- (vi) Hospital beds (D)
- (vii) Internal Generated revenue
- (viii) Equality
- (ix) Terrain
- (x) Rainfall proportion (3years) (D)
- (xi) Rainfall proportion (3years) (I)

Where 'D' stands for direct measure 'I' stands for inverse measure
 POPULATION

Local Government Data Entry : (program name:/population. prg)

The database files are: Number.dbf and locals.dbf. number dbf stores the number of local governments to be processed and has number as the field name. The detail of the structure of number .dbf is given below :

TABLE 1

<u>S/N</u>	<u>FIELD</u> <u>NAME</u>	<u>FIELD</u> <u>TYPE</u>	<u>DEC</u>	<u>INDEX</u>
1	Number	3	-	N

The input to the locals.dbf are name of state, local government. The design of the structure of locals.dbf is shown in table II .

TABLE II

S/N	FIELD NAME	FIELD TYPE	WIDTH	DEC	INDEX	REMARK
1	Local	Character	15	-	N	Locals Name
2	Lpop	Numeric	-	-	N	Local Pop.
3	Lindex	Numeric	-	-	N	Local Ind.
4	tlindex	Numeric	-	-	N	Total St. Ind.
5	Spop	Numeric		-	N	State

The format of lpopulation is shown below.

Number of Local Government:-----

Local Government: -----

Population: -----

State Government (program name: population.prg)

The database file are number .dbf . number dbf does the same work as in local government (lpopulation.prg) . The input to states.dbf are number of states, name of state and population of the states.

The structure of states.dbf is shown in table

TABLE III

S/N	FIELD NAME	FIELD TYPE	WIDTH	DEC.	INDEX	REMARK
1	State	Character	15	-	N	State's Name
2	Spop	Numeric	-	-	N	State's Pop.
3	Sindex	Numeric	-	-	N	State's Ind.
4	Lpop	Numeric	-	-	N	Country's pop
5	Tsindex	Numeric	-	-	N	Total Con. Ind.

Completion . Form Format

Number of state : -----

State: -----

Population: -----

Refer to pages 85 to 106 for formats and structures of other files .

Computation:

$$S_{pop} = S_{pop} + I_{pop}$$

$$L_{index} = (I_{pop} / S_{pop})$$

$$T_{index} = T_{index} + L_{index}$$

Refer to pages 51 to 74 for computation of other programs .

4.4 OUTPUT SPECIFICATION:

The out put of the program is directed to a text file named; Lland . out for Local Government while Sland . out for State Government. Refer to page 82 for the format of the output (form).

4.5 PROGRAM DESIGN

The system flow chart shown in the next page pictorially represent the captioning of information into the database files as well as deleting the records from the records from the files (text files) . It also allows the generating of reports either through the screen or the printer and quitting from the stem.

4.6 PROGRAM DEVELOPMENT

The program, project. prg is written in Turbo 6.0 and dbase iv tailored towards the result from investigation of the existing system of operation in the department. This program will replace the manual system of computing the revenue allocation indices.

4.7 PROGRAM DESCRIPTION AND MAIN PROCESS

The program is a menu-driven. At the running of the program, an introductory message is displayed followed by password screen . if the user entered the correct password, a menu is displayed on the screen consisting of three options as below :

- [1] Local Government
- [2] State Government
- [3] Quit To Dos Prompt

The above listed options are therefore sub-programs to the project. prg which call for another sub-programs. The format of the sub-menu called for is shown below :

DATA ENTRY

DELETE OUTPUT FILES (*.OUT)

REPORT

Upon execution of DATA ENTRY , another sub-menu is pulled out;

POPULATION

LAND MASS

SECONDARY SCHOOL ENROLMENT (D)

SECONDARY SCHOOL ENROLMENT (I)

PRIMARY SCHOOL ENROLMENT (D)

HOSPITAL BED

INTERNAL GENERATED REVENUE

EQUALITY

TERRAIN

RAINFALL PROPORTION (3 YEARS) (D)

RAINFALL PROPORTION (3YEARS) (I)

Refer to page 80 for details.

PROJECT .PRG

Project .prg is the main program file that displays the introduction screen, password screen, and menu options. Refer to page 46 to 48 and 76 for the main program, block chart respectively. The options provided are described below :

[1] DATA ENTRY :Choosing this option, the program will display sub-menu as shown in program development section. On clicking any of the system, the program will execute the appropriate module. This module will enable the user to input records and internal computation is carried out by the system . the data entry form will be presented to the user to enable him/her input records.

Refer to pages 48 and 80 for data entry sub modules and output respectively.

[2] DELETE OUTPUT FILE : At the beginning of the running of the program, a menu is displayed on the screen to enable the user to select the text file/record to delete. If the user clicks the text file he wanted to delete, the program will permanently remove the text file from the output files . If the user did not click the text file displayed, the text file remains in the output files.

Refer to pages 49 and 82 to delete programs and output respectively

[3] REPORT GENERATOR : This program module will enable the user to generate report from the file created in the course of data output. The program to generate report looks familiar with that of delete program. These input will either be obtained through the computer screen or the printer. To view or print output, the output device to be used must be selected after clicking the appropriate text file created in the course of data input/computation.

Refer to page 50 and 84 for the report/text file codes, report program and outputs respectively.

[4] QUIT : By selecting the quit option from the menu, the program will quit and pass control to the MS-dos programs.

Refer to page 46 and 79 for quit program and output respectively.

4.8 HOW TO RUN THE PROGRAM

(i) At The Dos Prompt :

Type DBASE and press ENTER key. Wait for some few seconds, DBASE IV will be loaded and present you with control centre . Press ESC key and letter 'Y' to change the environment from control centre to Dot prompt. Insert your program disk either in drive A: or B: and change default to the drive where your programs reside by typing the command below at the dot prompt.

SET DEFAULT TO A: OR B:

(ii) At The Dot Prompt :

Type DO PROJECT and press ENTER key. DBASE IV will execute the program. An introduction screen will be displayed and the user will be asked to press any key to continue . The program will also ask the user to enter password and if the password is correct , the main menu consisting of listed options will appear.

4.9 SYSTEM REQUIREMENTS

The following components are essential for implementation of the proposed system.

(A) Hardware Requirement

(i) IBM PC or Compatible

_ Pentium 166MHZ Speed or above

_ 16 MB RAM

_ 2.1 GB Hard Disk

_ 2 Serial/Parallel Cards

_ 4 Free Expansion Slots

_ 3.5" Floppy Disk Drive

_ Window 95 Keyboard

(ii) Monitor

_ 14" Colour SVGA

(iii) Printer

_ Laser Jet or Heavy Duty Epson Printer - LQ2 (1070)

(iv) Power Storage

_ APC Smart non-interruptible power supply (UPS) 500 VA

(B) Software Requirement

_ MS-DOS Version 6.22/Window'95

_ DBASE iv and Turbo Pascal 6.0

(C) **Consumable Items**

_ 3.52 HD Diskettes (Maxwell)

_ Maxwell Ribbons

_ Computer Papers

4.10 **TRAINING NEEDS**

The computer professionals in the Planning, Research and Statistics department of the Commission are capable of manning the proposed system when operational but, an in-house training would need to be given to some staff in Allocation department to refresh their knowledge on how to use the project. Prg and other computer operation.

4.11 **SYSTEM CONVERSION**

The system conversion/changeover may be achieved in a number of ways; The most common methods are : Direct, Parallel Running, Pilot Running and Staged Changeover.

- (a) **Direct Changeover** :- This method is the complete replacement of the old system by the new, in one move . it is a bold move, which should be undertaken only when everyone concerned has confidence in the new system. When a direct changeover is planned, system test and training should be comprehensive, and the changeover itself planned in detail. This method is potentially the least expensive but, the most risky. For security reasons, the old system may be held in abeyance including people and equipment . In the event of a major failure of the new system the organisation would revert to the old system .

- (b) **Parallel Running** :- This means processing current data by both the old and the new systems to cross-check the results. Its main attraction is that the old system is kept alive and operational until the new system has been proved for at least one system cycle, using full live data in the real operational environment of a place, people, equipment and time. It allows the results of the new system to be compared with the old system before acceptance by the user, thereby promoting user confidence. Its main disadvantage is the extra cost the difficulty and impracticability of user staff having to carry out the different clerical operations for two systems on the time available for one .
- (c) **Pilot Running** :- This is similar in concept to parallel running . Data from one or more previous period for the whole or part of the system is run on the new system after results have been obtained from the old system, and the new results are compared with the old ones.
- (d) **Staged Changeover** :- This involves a series of limited size direct changeover, the new system being introduced piece by piece . A complete part or logical section, is committed to the new system while the remaining parts or sections are processed by the old system. Only when the selected part is operating satisfactorily is the remainder transferred. This method reduces the risks inherent in a direct changeover of the whole system and enables the analyst and users to learn from mistakes made as the changeover progress. But on the other hand, it creates problems of controlling the selected parts of the old and new system and it tends to prolong the implementation period.

For the newly designed system, the recommended system of conversion to be used is PILOT CHANGEOVER in view of the fact that it is not as destructive as parallel operation, since timing is less critical. This method is more practical form of change over for organisational reasons.

4.12 REVIEW AND MAINTENANCE

Once the system has become operational, there is a need to examine it to see if it has met its objectives particularly in terms of cost and benefits. The system will also need to be reviewed and periodically due to the following reasons :

- (i) To deal with unforeseen problems arising in operation
- (ii) To confirm that planned objectives are being met and to take action if they are not
- (iii) Adaptability of the software to the environment must also be ensured.
The users also must be able to cope with the new system.

Maintenance of the software will normally be the responsibility of the programmer, who ensures that the package is always in good working order or condition and necessary precautions are taken to uphold the integrity of the program.

CHAPTER :FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 SUMMARY

In conformity with the objective of this project, we have been able to convert the manual system of computing the revenue allocation indices for state and local governments to a computerised system. There has been an in depth—study of the existing system and the need for computerisation . The reasons which have been mentioned in the various chapters.

The analysis of the existing system pointed out the problem being experienced, the causes of the problems and the need for correction or improvement on the system. The design of the new system was done based on the users requirement and this includes computer program for running the system.

Finally, the implementation of the designed system to be used by the Commission includes the training of the existing staff, hard ware and soft ware requirements, method of conversion and review/maintenance.

5.2 CONCLUSION

This project has been tested and found to function effectively to solve the problems of data manipulation and security of records. It also allows storage of records in the computer system so as to eliminate the use of files cabinets that occupies much space of the offices. And there will be better management of data and information with high degree of accuracy and speed of response.

5.3 RECOMMENDATIONS

It is there fore recommended that;

- (1) The National Revenue Mobilisation Allocation and Fiscal Commission should computerised it's Revenue Allocation indices computation.

- (2) Local Area Net Work System should be used in installing the computers in all the divisions within the departments, so as to facilitate easy communication of information and retrieval.
- (3) The Commission should also go into wide Area Network with a view to linking all the seven zonal offices located in Kano, Port-Harcourt, Enugu, Akure, Yola, jos and Lagos. This is to allow communication of information between the headquarters and the zonal offices.
- (4) A qualified system Analyst/programmer should oversee the central computer system with a view to assessing the performance of the new system and to make continuous adjustments as demand arises. This is to ensure that the system achieves its goals.
- (5) Since full automation of allocation indices computation is suggested, the existing staff in the Allocation Department need to be given a refresher training to afford them new skills and develop themselves gradually.

5 PROGRAM

```

{*****}
*** Name:JOHN ANDA ***
*** Registration No: PGD/MCS/97/304 ***
{*****}

```

```
Program Vector(Input, Output);
```

```
Uses Crt, Printer, Graph;
```

```
Const
```

```
{User define fill pattern}
```

```
User1 : FillPatternType = ($AA, $59, $AA, $55, $AA, $55, $AA, $55);
```

```
User2 : FillPatternType = ($FF, $10, $FF, $10, $FF, $10, $FF, $10);
```

```
User3 : FillPatternType = ($AE, $A4, $AE, $A4, $AE, $A4, $AE, $A4);
```

```
Var
```

```
Graphdriver:integer;
```

```
GraphMode:integer;
```

```
OldExitProc:pointer;
```

```
MaxX,MaxY:word;
```

```
ErrorCode:integer;
```

```
MaxColor:word;
```

```
I:integer;
```

```
Keyp:String[1];
```

```
{F+}
```

```
procedure MyExitProc;
```

```
begin
```

```
ExitProc := OldExitProc; { Restore exit procedure address }
```

```
CloseGraph; { Shut down the graphics system }
```

```
end; { MyExitProc }
```

```
{F-}
```

```
Procedure Initialize;
```

```
{ Initialize graphics and report any errors that may occur }
```

```
var
```

```
InGraphicsMode : boolean; { Flags initialization of graphics mode }
```

```
PathToDriver : string; { Stores the DOS path to *.BGI & *.CHR }
```

```
begin
```

```
{ when using Crt and graphics, turn off Crt's memory-mapped writes }
```

```
DirectVideo := False;
```

```
OldExitProc := ExitProc; { save previous exit proc }
```

```
PathToDriver := 'C:\TP\BGI';
```

```
repeat
```

```
{ $IFDEF Use8514 }
```

```
{ check for Use8514 $DEFINE }
```

```
GraphDriver := IBM8514;
```

```
GraphMode := IBM8514Hi;
```

```
{ $ELSE }
```

```
GraphDriver := Detect;
```

```
{ use autodetection }
```

```
{ $ENDIF }
```

```
InitGraph(GraphDriver, GraphMode, PathToDriver);
```

```
ErrorCode := GraphResult; { preserve error return }
```

```
if ErrorCode <> grOK then { error? }
```

```
begin
```

```
Writeln('Graphics error: ', GraphErrorMsg(ErrorCode));
```

```
if ErrorCode = grFileNotFound then { Can't find driver file }
```

```
begin
```

```
Writeln('Enter full path to BGI driver or type <Ctrl-Break> to quit:
```

```
Readln(PathToDriver);
```

```
Writeln;
```

```
end
```

```

    else
        Halt(1); { Some other error: terminate }
    end;
until ErrorCode = grOK;
Randomize; { init random number generator }
MaxColor := GetMaxColor; { Get the maximum allowable drawing color }
MaxX := GetMaxX; { Get screen resolution values }
MaxY := GetMaxY;
end; { Initialize }

```

```

function Int2Str(L : LongInt) : string;
{ Converts an integer to a string for use with OutText, OutTextXY }
var
    S : string;
begin
    Str(L, S);
    Int2Str := S;
end; { Int2Str }

```

```

Procedure box(x1,y1,x2,y2,r1,r2,c1,c2,c3:integer);
{Draw box with given parameters}
begin
    setcolor(c1);
    setlinestyle(Solidln,0,Normwidth);
    arc(x1+r1,y1+r1,90,180,r1);
    arc(x2-r2,y1+r2,0,90,r2);
    arc(x1+r2,y2-r2,180,270,r2);
    arc(x2-r1,y2-r1,270,0,r1);
    line(x1+r1,y1,x2-r2,y1);
    line(x2,y1+r2,x2,y2-r1);
    line(x2-r1,y2,x1+r2,y2);
    line(x1,y2-r2,x1,y1+r1);
    { setfillpattern(user2,c1); }
    floodfill(x1+10,y1+10,c1);
    setlinestyle(Solidln,0,Thickwidth);
    setcolor(c2);
    arc(x1+r1,y1+r1,90,180,r1);
    arc(x2-r2,y1+r2,0,90,r2);
    setcolor(c3);
    arc(x1+r2,y2-r2,180,270,r2);
    arc(x2-r1,y2-r1,270,0,r1);
    setcolor(c2);
    line(x1+r1,y1,x2-r2,y1);
    line(x2,y1+r2,x2,y2-r1);
    setcolor(c3);
    line(x2-r1,y2,x1+r2,y2);
    line(x1,y2-r2,x1,y1+r1);
end;

```

```

Procedure Sector(x1,y1,st,ed,xr,yr:Integer);
Begin
    Sector(x1,y1,st,ed,xr,yr);
    line(x1+xr,y1,x1,y1-yr);
    line(x1,y1-yr,x1-xr,y1);
    line(x1,y1,x1,y1-yr);
End;

```

```

Procedure Press;
{Press any key to Continue}
Begin
    Setcolor(Yellow);
    Settextstyle(Defaultfont, Horizdir,2);

```

```
Outtextxy(100, 450, '- Press any key to continue');
Keyp := Readkey;
End;
```

```
Procedure Background;
{Display Background}
Begin;
  initialize;
  setfillpattern(user1,2);
  box(5,10,Maxx-10,Maxy-10,20,20,2,15,15);
  secter(50,50,180,360,20,20);
  secter(590,50,180,360,20,20);
  secter(50,420,180,360,20,20);
  secter(590,420,180,360,20,20);
End;
```

```
Procedure Page2;
Begin
  Background;
  settextstyle(Defaultfont, Horizdir,4);
  Setcolor(Yellow);
  Outtextxy(50, 100, 'COMPUTERISATION');
  Outtextxy(260, 190, 'OF');
  Outtextxy(180, 280, 'INDICES');
  Setcolor(Blue);
  Keyp := Readkey;
  Press;
  for i := 1 to 20 do
  Begin
    delay(1500);
    Sound(1238);
    box(15*i,10*i,Maxx-(15*i),Maxy-(10*i),20,20,4,15,15);
    NoSound;
  End;
  for i := 20 downto 1 do
  Begin
    delay(500);
    bar(15*i,10*i,Maxx-(15*i),Maxy-(10*i));
  End;
End;
```

```
Procedure Page1;
Begin
  Background;
  settextstyle(Defaultfont, Horizdir,2);
  Setcolor(blue);
  Outtextxy(100, 50, 'COMPUTERISATION OF INDICES ');
  Outtextxy(100, 90, 'USE IN REVENUE ALLOCATION');
  Outtextxy(135, 130, '"A CASE STUDY OF ');
  Outtextxy(50, 170, ' NATIONAL REVENUE MOBILIZATION ');
  Outtextxy(40, 210, 'ALLOCATION AND FISCAL COMMISSION" ');
  Outtextxy(270, 250, 'BY');
  Settextstyle(Defaultfont, Horizdir,2);
  Outtextxy(200, 280, 'JOHN ANDA');
  Outtextxy(180, 330, 'PGD/MCS/97/304');
  Settextstyle(Defaultfont, Horizdir,2);
  Outtextxy(30, 370, 'F.U.T. Minna (COMPUTER & MATHS DEPT.)');
  Press;
  for i := 20 downto 1 do
  Begin
    delay(500);
    bar(15*i,10*i,Maxx-(15*i),Maxy-(10*i));
  End;
End;
```

End;

BEGIN {Main Program}

Page1;

Page2;

CloseGraph;

END.

```
RUN A:INTRO
DO APASS
DO HEADER
DO MAIN_DEF
ACTIVATE POPUP MAINMENU
```

```
PROCEDURE MAIN_DEF
DEFINE POPUP MAINMENU FROM 10,30 TO 15,60 ;
MESSAGE "PRESS FIRST LETTER OF MENU CHOICE, OR HIGHLIGHT AND PRESS <ENTER>"
DEFINE BAR 1 OF MAINMENU PROMPT "===MAIN MENU==="SKIP
DEFINE BAR 2 OF MAINMENU PROMPT "LOCAL GOVERNMENT"
DEFINE BAR 3 OF MAINMENU PROMPT "STATE GOVERNMENT"
DEFINE BAR 4 OF MAINMENU PROMPT "QUIT TO DOS PROMPT"
ON SELECTION POPUP MAINMENU DO MAIN
```

```
RETURN
```

```
PROCEDURE MAIN
DO CASE
CASE BAR() = 2
DO LOCAL_DEF
ACTIVATE POPUP LOCAL_MENU
```

```
CASE BAR() = 3
DO STATE_DEF
ACTIVATE POPUP STATE_MENU
```

```
CASE BAR() = 4
QUIT
ENDCASE
RETURN
```

```
PROCEDURE STATE_DEF
DEFINE POPUP STATE_MENU FROM 10,30 TO 15,60;
MESSAGE "PRESS FIRST LETTER OF MENU CHOICE, OR HIGHLIGHT AND PRESS <ENTER>"
DEFINE BAR 1 OF STATE_MENU PROMPT "===STATE GOVERNMENT MENU==="SKIP
DEFINE BAR 2 OF STATE_MENU PROMPT "DATA ENTRY"
DEFINE BAR 3 OF STATE_MENU PROMPT "DELETE OUTPUT FILES (*.OUT)"
DEFINE BAR 4 OF STATE_MENU PROMPT "REPORT"
DEFINE BAR 5 OF STATE_MENU PROMPT "EXIT"
ON SELECTION POPUP STATE_MENU DO STATEE
```

```
RETURN
```

```
PROCEDURE STATEE
DO CASE
CASE BAR() = 2
DO DATAS_DEF
ACTIVATE POPUP DATAS_MENU
```

```
CASE BAR() = 3
DO DELETETL
CASE BAR() = 4
DO REPORTL
CASE BAR() = 5
RETURN
ENDCASE
```

```
RETURN
```

```
PROCEDURE DATAS_DEF
CLEAR
DO HEADER
DEFINE POPUP DATAS_MENU FROM 7,20 TO 22,70 ;
```

```

MESSAGE "PRESS FIRST LETTER OF MENU CHOICE ,OR HIGHLIGHT AND PRESS <ENTER>"
DEFINE BAR 1 OF DATAS_MENU PROMPT "==STATE GOVERNMENT DATA ENTRY MENU==" SK
DEFINE BAR 2 OF DATAS_MENU PROMPT "POPULATION"
DEFINE BAR 3 OF DATAS_MENU PROMPT "LAND MASS"
DEFINE BAR 4 OF DATAS_MENU PROMPT "SECONDARY SCHOOL ENROLMENT(D) "
DEFINE BAR 5 OF DATAS_MENU PROMPT "SECONDARY SCHOOL ENROLMENT(I) "
DEFINE BAR 6 OF DATAS_MENU PROMPT "PRIMARY SCHOOL ENROLMENT(D) "
DEFINE BAR 7 OF DATAS_MENU PROMPT "HOSPITAL BEDS (D) "
DEFINE BAR 8 OF DATAS_MENU PROMPT "INTERNAL GENERATED REVENUE (D) "
DEFINE BAR 9 OF DATAS_MENU PROMPT "EQUALITY (D) "
DEFINE BAR 10 OF DATAS_MENU PROMPT "TERRAIN(D) "
DEFINE BAR 11 OF DATAS_MENU PROMPT "RAINFALL PROPORTIONA(3 YEARS) (D) "
DEFINE BAR 12 OF DATAS_MENU PROMPT "RAINFALL PROPORTIONA(3 YEARS) (I) "

```

ON SELECTION POPUP DATAS_MENU DO STATE

RETURN

PROCEDURE STATE

```

DO CASE
  CASE BAR() = 2
    DO POPULATION
    WAIT ' '
    CLEAR
  CASE BAR() = 3
    DO LAND
    WAIT ' '
    CLEAR
  CASE BAR() = 4
    DO ENROLMENT
    WAIT ' '
    CLEAR
  CASE BAR() = 5
    DO IENROLMENT
    WAIT ' '
    CLEAR
  CASE BAR() = 6
    DO PENROLMENT
    WAIT ' '
    CLEAR
  CASE BAR() = 7
    DO SHB
    WAIT ' '
    CLEAR
  CASE BAR() = 8
    DO REV
    WAIT ' '
    CLEAR
  CASE BAR() = 9
    DO EQUAL
    WAIT ' '
    CLEAR
  CASE BAR() = 10
    DO TERRIAN
    WAIT ' '
    CLEAR
  CASE BAR() = 11
    DO RAIN
    WAIT ' '
    CLEAR
  CASE BAR() = 12
    DO IRAIN
    WAIT ' '
    CLEAR

```

ENDCASE

RETURN

PROCEDURE LOCAL_DEF

```
DEFINE POPUP LOCAL_MENU FROM 10,30 TO 15,60 ;
MESSAGE "PRESS FIRST LETTER OF MENU CHOICE, OR HIGHLIGHT AND PRESS <ENTER>"
DEFINE BAR 1 OF LOCAL_MENU PROMPT "===LOCAL GOVERNMENT MENU=== "SKIP
DEFINE BAR 2 OF LOCAL_MENU PROMPT "DATA ENTRY"
DEFINE BAR 3 OF LOCAL_MENU PROMPT "DELETE OUTPUT FILES (*.OUT) "
DEFINE BAR 4 OF LOCAL_MENU PROMPT "REPORT"
DEFINE BAR 5 OF LOCAL_MENU PROMPT "EXIT"
ON SELECTION POPUP LOCAL_MENU DO LOCALL
```

RETURN

PROCEDURE LOCALL

```
DO CASE
CASE BAR() = 2
DO DATAL_DEF
ACTIVATE POPUP DATAL_MENU
```

```
CASE BAR() = 3
DO DELETTEL
CASE BAR() = 4
DO REPORTL
CASE BAR() = 5
RETURN
ENDCASE
```

RETURN

PROCEDURE DATAL_DEF

```
CLEAR
DO HEADER
DEFINE POPUP DATAL_MENU FROM 7,20 TO 22,70 ;
MESSAGE "PRESS FIRST LETTER OF MENU CHOICE ,OR HIGHLIGHT AND PRESS <ENTER>:"
DEFINE BAR 1 OF DATAL_MENU PROMPT "===LOCAL GOVERNMENT DATA ENTRY MENU=== " SI
DEFINE BAR 2 OF DATAL_MENU PROMPT "POPULATION"
DEFINE BAR 3 OF DATAL_MENU PROMPT "LAND MASS"
DEFINE BAR 4 OF DATAL_MENU PROMPT "SECONDARY SCHOOL ENROLMENT (D) "
DEFINE BAR 5 OF DATAL_MENU PROMPT "SECONDARY SCHOOL ENROLMENT (I) "
DEFINE BAR 6 OF DATAL_MENU PROMPT "PRIMARY SCHOOL ENROLMENT (D) "
DEFINE BAR 7 OF DATAL_MENU PROMPT "HOSPITAL BEDS (D) "
DEFINE BAR 8 OF DATAL_MENU PROMPT "INTERNAL GENERATED REVENUE (D) "
DEFINE BAR 9 OF DATAL_MENU PROMPT "EQUALITY (D) "
DEFINE BAR 10 OF DATAL_MENU PROMPT "TERRAIN (D) "
DEFINE BAR 11 OF DATAL_MENU PROMPT "RAINFALL PROPORTIONA (3 YEARS) (D) "
DEFINE BAR 12 OF DATAL_MENU PROMPT "RAINFALL PROPORTIONA (3 YEARS) (I) "
```

```
ON SELECTION POPUP DATAL_MENU DO LOCAL
```

RETURN

PROCEDURE LOCAL

```
DO CASE
CASE BAR() = 2
DO LPOPULATION
WAIT ' '
CLEAR
CASE BAR() = 3
```

```
DO LLAND
WAIT ' '
CLEAR
CASE BAR() = 4
DO LENROLMENT
WAIT ' '
CLEAR
CASE BAR() = 5
DO LIENROLMENT
WAIT ' '
CLEAR
CASE BAR() = 6
DO LPENROLMENT
WAIT ' '
CLEAR
CASE BAR() = 7
DO LSHB
WAIT ' '
CLEAR
CASE BAR() = 8
DO LREV
WAIT ' '
CLEAR
CASE BAR() = 9
DO LEQUAL
WAIT ' '
CLEAR
CASE BAR() = 10
DO LTERRIAN
WAIT ' '
CLEAR
CASE BAR() = 11
DO LRAIN
WAIT ' '
CLEAR
CASE BAR() = 12
DO LIRAIN
WAIT ' '
CLEAR
```

ENDCASE

RETURN

PROCEDURE DELETTEL

```
CLEAR
DO HEADER
DEFINE POPUP FILESPOP FROM 9,30 TO 16,60 PROMPT FILES LIKE A:*.OUT ;
MESSAGE "PRESS <ENTER> TO SELECT A FILE TO DELETE"
ON SELECTION POPUP FILESPOP DO DELE_ACTION
ACTIVATE POPUP FILESPOP
RETURN
```

PROCEDURE DELE_ACTION

```
STORE SPACE(12) TO MFILE
MFILE = PROMPT()
IF MFILE = " "
@21,30 SAY " NO FILE SELECTED"
WAIT "PRESS ANY KEY TO RETURN"
```

RETURN
ENDIF

```
ERASE &MFILE
WAIT "FILE HAS BEEN DELETED"
DEACTIVATE POPUP
RELEASE POPUP FILESPOP
RETURN
```

```
PROCEDURE REPORTL
```

```
  CLEAR
  DO HEADER
  DEFINE POPUP REPLPOP FROM 9,30 TO 16,60 PROMPT FILES LIKE A:*.OUT ;
  MESSAGE "PRESS <ENTER> TO SELECT A OUTPUT FILE"
  ON SELECTION POPUP REPLPOP DO REP_ACTION
  ACTIVATE POPUP REPLPOP
RETURN
```

```
PROCEDURE REP_ACTION
```

```
  STORE SPACE(12) TO MFILE
  MFILE = PROMPT()
  IF MFILE = " "
    @21,30 SAY "NO OUTPUT FILE SELECTED"
    WAIT "PRESS ANY KEY TO RETURN"
  RETURN
  ENDIF
  CLOSE ALL
  STORE SPACE(1) TO REPLY
  @22,30 SAY "DESTINATION :PRINTER OR SCREEN (P/S) " GET REPLY
  READ
  IF REPLY = "P" .OR. REPLY = "p"
    SET PRINTER ON
    RUN TYPE &MFILE >PRN
    SET PRINTER OFF
  ELSE
    IF REPLY = "S" .OR. REPLY = "s"
      CLEAR
      TYPE &MFILE
      WAIT " "
    ENDIF
  ENDIF
  DEACTIVATE POPUP
  RELEASE POPUP REPLPOP
RETURN
```

```
*POPULATION
procedure population
set alternate to "A:spop.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
append blank
number = mnumber
use states
declare state[number], spop[number], sindex[number]
k=1
cpop = 0
tsindex = 0
```

```

do while k <= number
  store space(15) to state[k]
  spop[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@"
  @13,5 say "population:" get spop[k] pict "9999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace cpop with (cpop + spop[k])
  k = k + 1
enddo
k = 1
do while k <= number
  sindex[k] = (spop[k]/cpop)*100
  replace tsindex with (tsindex + sindex[k])
  k = k + 1
enddo
clear
k= 1
? space(5) + " S/NO      STATE                POPULATION                INDEX      "
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|', state[k], ' |      ', str(spop[k],10), ' |      ', str(sindex[k],6,2)

  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(47) + "TOTAL =>" , ROUND(tsindex,1)

set alternate off
close data
return

```

```

procedure land
*LAND MASS
set alternate to "A:sland.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use statel
declare state[number],sland[number],siindex[number]
k=1
cland = 0
tsiindex = 0
do while k <= number
  store space(15) to state[k]
  sland[k] = 0

```

```

k = k + 1
enddo
k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@"
  @13,5 say "state land mass:" get sland[k] pict "9999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace cland with (cland + sland[k])
  k = k + 1
enddo
k = 1
do while k <= number
  siindex[k] = (sland[k]/cland)*100
  replace tsiindex with (tsiindex + siindex[k])
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO    STATE          LAND MASS          INDEX          "
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|',state[k], ' |          ',str(sland[K],10), ' |          ',str(siindex[K],6,2)

  k = k + 1

enddo
?space(5), replicate ('=',60)
? space(47) + "TOTAL =>" ,ROUND(tsiindex,1)

set alternate off
close data

return

```

```

*SECONDARY SCHOOL ENROLMENT
procedure enrolment
set alternate to "A:sse.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use statee
declare state[number], sse[number], seindex[number]
k=1
csse = 0
tseindex = 0
do while k <= number
  store space(15) to state[k]
  sse[k] = 0
  k = k + 1
enddo

```

```

k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@!"
  @13,5 say "secondary school enrolment:" get sse[k] pict "999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace csse with (csse + sse[k])
  k = k + 1
enddo
k = 1
do while k <= number
  seindex[k] = (sse[k]/csse)*100
  replace tseindex with (tseindex + seindex[k])
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO STATE SECONDARY ENROLMENT INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|', state[k], ' | ', str(sse[k],10), ' | ', str(seindex[k],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tseindex,1)
set alternate off
close data
return

```

*PRIMARY SCHOOL ENROLMENT

```

procedure penrolment
set alternate to 'A:spse.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use statepe
declare state[number], pse[number], speindex[number]
k=1
cpse = 0
tspeindex = 0
do while k <= number
  store space(15) to state[k]
  pse[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@!"
  @13,5 say "primary school enrolment:" get pse[k] pict "999999999"
  read
  k = k + 1

```

```

enddo
k = 1
do while k <= number
  replace cpse with (cpse + pse[k])
  k = k + 1
enddo
k = 1
do while k <= number
  speindex[k] = (pse[k]/cpse)*100
  replace tspeindex with (tspeindex + speindex[k])
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO STATE PRIMARY ENROLMENT INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',state[k],' | ',str(pse[k],10),' | ',str(speindex[k],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tspeindex,1)
set alternate off
close data
return

```

*HOSPITAL BEDS

```

procedure shb
set alternate to 'A:shb.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use statehb
declare state[number], shb[number], shbindex[number]
k=1
chb = 0
tshbindex = 0
do while k <= number
  store space(15) to state[k]
  shb[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@"
  @13,5 say "state hospital bed:" get shb[k] pict "9999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace chb with (chb + shb[k])

```

```

    k = k + 1
enddo
k = 1
do while k <= number
    shbindex[k] = (shb[k]/chb)*100
    replace tshbindex with (tshbindex + shbindex[k])
    k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO      STATE      HOSPITAL BED      INDEX  "
?space(5), replicate ('=',60)
do while k <= number
    ?k, '|', state[k], ' |      ', str(shb[k],10), '      |      ', str(shbindex[k],6,2)
    k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tshbindex,1)
set alternate off
close data
return

```

*INTERNAL GENERATED REV.

```

procedure rev
set alternate to 'a:sigr.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use stater
declare state[number], year [number,2], rev [number,2], diff [number], srindex [number]
k=1
tdiff = 0
tperdiff = 0
tsrindex = 0
do while k <= number
    store space(15) to state[k]
    year[k,1]= 0
    year[k,2] = 0
    rev[k,1] = 0
    rev[k,2] = 0
    k = k + 1
enddo
k = 1
do while k <= number
    @10,5 say "state:" get state[k] pict "@"
    @12,5 say "first year:" get year[k,1] pict "9999"
    @14,5 say "second year:" get year[k,2] pict "9999"
    @16,5 say "first revenue:" get rev[k,1] pict "9999999999"
    @18,5 say "second revenue:" get rev[k,2] pict "9999999999"

    read
    k = k + 1
enddo
k = 1
do while k <= number

```

```

if rev[k,1] > rev[k,2]
  diff[k]= 0
else
  diff[k] = rev[k,2] - rev[k,1]
endif
replace tdiff with (tdiff + diff[k])
k = k + 1
enddo
k = 1
do while k <= number
  perdiff[k] = (diff[k]/rev[k,1])*100
  replace tperdiff with (tperdiff + perdiff[k])
  k = k + 1
enddo
clear
k = 1
do while k <= number
  srindex[k] = (perdiff[k]/tperdiff)*100
  replace tperdiff with (tperdiff + perdiff[k])
  replace tsrindex with (tsrindex + srindex[k])
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO STATE 1_REV | 2_REV |"
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|', state[k], ' | ', str(rev[k,1],10), ' | ', str(rev[k,2],10), ' | ', st
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tsrindex,1)
set alternate off
close data
return

```

```

*TERRIAN
procedure terrian
set alternate to 'A:st.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use statet
declare state[number], rs[number], rl[number], rh[number]
declare ws[number], wl[number], wh[number], tw[number], i[number]
k=1
gt = 0
gti=0
do while k <= number
  store space(15) to state[k]
  rs[k] = 0
  rl[k] = 0
  rh[k] = 0
  k = k + 1
enddo

```

```

k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@"
  @13,5 say "areal proportion of wetlands:" get rs[k] pict "9999999999"
  @15,5 say "areal proportion of plains:" get rl[k] pict "9999999999"
  @17,5 say "areal proportion of highland:" get rh[k] pict "9999999999"

  read
  k = k + 1
enddo
k = 1
do while k <= number
  ws[k] = rs[k] * 2.0
  wl[k] = rl[k] * 1.0
  wh[k] = rh[k] * 1.5
  tw[k] = ws[k] + wl[k] + wh[k]
  replace gt with (gt + tw[k] )
  k = k + 1
enddo
k = 1
do while k <= number
  i[k] = (tw[k]/gt)*100
  replace gti with (gti + i[k])
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO | STATE | ALL TERRIAN | INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',state[k],' | ',str(tw[K],10),' | ',str(i[K],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(gti,1)
set alternate off
close data
return

```

```

*EQUALITY
procedure equal
set alternate to 'A:sq.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@11,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use stateq
declare state[number]
k=1
tei = 0
do while k <= number
  store space(15) to state[k]

  k = k + 1
enddo

```

```

k = 1
do while k <= number
  @13,5 say "state:" get state[k] pict "@!"
  read
  k = k + 1
enddo

```

```

k = 1
do while k <= number

  replace index with (1.0/number)*100
  replace tei with (tei + index)
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO | STATE | INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',state[k],' | ',str(index,6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tei,1)
set alternate off
close data
return

```

*RAINFALL PROPORTION

```

procedure rain
set alternate to 'A:sr.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use stata
declare state[number], ar[number], rp[number]
k=1
tar = 0
trp = 0
do while k <= number
  store space(15) to state[k]
  ar[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@!"
  @13,5 say "average rainfall distribution (3 years):" get ar[k] pict "999999"
  read
  k = k + 1
enddo
k = 1

do while k <= number

```

```

replace tar with (tar + ar[k])
  k = k + 1
enddo
k = 1
do while k <= number
  rp[k] = (ar[k]/tar)*100
  replace trp with (trp +rp[k])
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO | STATE | AVERAGE | PROP.RATIO"
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',state[k],' | ',str(ar[k],10),' | ',str(rp[k],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(trp,1)
set alternate off
close data
return

```

*INVERSE SECONDARY ENROLMENT

```

procedure ienrolment
set alternate to 'A:isse.out'
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@9,5 say "number of state:" get mnumber pict "999"
read
number = mnumber
use stateis
declare state[number], sse[number], inverse[number], sdindex[number]
k=1
tinverse = 0
tsdindex = 0
do while k <= number
  store space(15) to state[k]
  sse[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @11,5 say "state:" get state[k] pict "@!"
  @13,5 say "secondary school enrolment:" get sse[k] pict "9999999999"
  read
  k = k + 1
enddo

k = 1
do while k <= number

  inverse[k] = (1/sse[k])

  k = k + 1

```

```
enddo
```

```
k = 1  
do while k <= number  
  replace tinverse with (tinverse + inverse[k])  
  k = k + 1  
enddo
```

```
k = 1  
do while k <= number  
  sdindex[k] = (inverse[k]/tinverse)*100  
  k = k + 1  
enddo
```

```
k = 1  
do while k <= number  
  replace tsdindex with (tsdindex + sdindex[k])  
  k = k + 1  
enddo
```

```
clear  
k= 1  
?space(5) + "S/NO | STATE | SEC.ENROL.(INVERSE) | INDEX "  
?space(5), replicate ('=',60)  
do while k <= number  
  ?k,'|',state[k],' | ',str(inverse[k],11,4),' | ',str(sdindex[k],11,4)  
  k = k + 1  
enddo  
?space(5), replicate ('=',60)  
? space(41) + "TOTAL=>" ,ROUND(tsdindex,1)  
set alternate off  
close data  
return
```

```
*INVERSE RAINFALL PROPORTION
```

```
procedure irain  
set alternate to 'A:isr.out'  
set alternate on  
set talk off  
clear  
do header  
use number  
store 0 to mnumber  
@9,5 say "number of state:" get mnumber pict "999"  
read  
number = mnumber  
use stateir  
declare state[number], arv[number], iinverse[number], sdiindex[number]  
k=1  
tiinverse = 0  
tsdiindex = 0  
do while k <= number  
  store space(15) to state[k]  
  arv[k] = 0  
  k = k + 1  
enddo  
k = 1  
do while k <= number
```

```

@11,5 say "state:" get state[k] pict "@"
@13,5 say "average rainfall over(3 years):" get arv[k] pict "9999999999"
read
k = k + 1
enddo

k = 1
do while k <= number

  iinverse[k] = (1/arv[k])

  k = k + 1
enddo

k = 1
do while k <= number
  replace tiinverse with (tiinverse + iinverse[k])
  k = k + 1
enddo

k = 1
do while k <= number
  sdiindex[k] = (iinverse[k]/tiinverse)*100
  k = k + 1
enddo

k = 1
do while k <= number
  replace tsdiindex with (tsdiindex + sdiindex[k] )
  k = k + 1
enddo
clear
k= 1
?space(5) + "S/NO |          STATE | AVEG.RAINFALL(INVERSE) | INDEX "
?
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',state[k],' |          ',str(iinverse[k],11,4),' |          ',str(sdiindex[k],11,4)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tsdiindex,1)
set alternate off
close data
return

```

*LOCAL GOVERNMENT POPULATION

```

procedure lpopulation
set talk off
set alternate to 'A:lpop.out'
set alternate on
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read

```

```

append blank
number = mnumber
use locals
declare local [number], lpop [number], lindex [number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

k=1
do while k <= number
  store space(15) to local[k]
  lpop[k] = 0
  k = k + 1
enddo

k = 1
tlindex = 0
spop = 0
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  @15,5 say "population:" get lpop[k] pict "9999999999"
  read

  k = k + 1
enddo

k=1
do while k <= number
  replace spop with (spop + lpop[k])
  k = k + 1
enddo

k=1
do while k <= number
  lindex[k] = (lpop[k]/spop)*100

  replace tlindex with (tlindex + lindex[k])
  k = k + 1
enddo

k=1
clear
?space(20), "POPULATION FACTOR FOR", state
? space(5) + " S/NO LOCAL GOVERT. POPULATION INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|', local[k], ' | ', str(lpop[k],10), ' | ', str(lindex[k],6,2)

  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(47) + "TOTAL =>" , ROUND(tlindex,1)

set alternate off
close data
return

```

```

procedure lland
*LOCAL GOVERNMENT LAND MASS
set talk off
set alternate to "A:lland.out"
set alternate on
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use local1
declare local[number],lland[number],liindex[number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

k=1
tliindex = 0
sland = 0
do while k <= number
  store space(15) to local[k]
  lland[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@!"
  @15,5 say "local government land mass:" get lland[k] pict "9999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace sland with (sland + lland[k])
  k = k + 1
enddo
k = 1
do while k <= number
  liindex[k] = (lland[k]/sland)*100
  replace tliindex with (tliindex + liindex[k])
  k = k + 1
enddo
clear
k= 1
?space(20),"LAND MASS FACTOR FOR", state
?space(5) + "S/NO LOCAL GOVERN. LAND MASS INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',local[k],' | ',str(lland[k],10),' | ',str(liindex[k],6,2)

  k = k + 1

enddo
?space(5), replicate ('=',60)
? space(47) + "TOTAL =>" ,ROUND(tliindex,1)

set alternate off
close data

```

return

```
*LOCAL GOVERNMENT SECONDARY SCHOOL ENROLMENT
procedure lenrolment
set alternate to "A:lse.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use locale
declare local[number],lse[number],leindex[number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

k=1
tleindex = 0
ssse = 0
do while k <= number
  store space(15) to local[k]
  lse[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  @15,5 say "secondary school enrolment:" get lse[k] pict "9999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace ssse with (ssse + lse[k])
  k = k + 1
enddo
k = 1
do while k <= number
  leindex[k] = (lse[k]/ssse)*100
  replace tleindex with (tleindex + leindex[k])
  k = k + 1
enddo
clear
k= 1
?space(20),"SECONDARY SCHOOL ENROLNMENT FOR", state
?space(5) + "S/NO LOCAL GOVERN. SECONDARY ENROLMENT INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',local[k],' | ',str(lse[K],10),' | ',str(leindex[K],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,tleindex
set alternate off
close data
```

return

*LOCAL GOVERNMENT PRIMARY SCHOOL ENROLMENT

procedure lpenrolment

set talk off

set alternate to "A:lpse.out"

set alternate off

clear

do header

use number

store 0 to mnumber

@11,5 say "number of local government:" get mnumber pict "999"

read

number = mnumber

use localpe

declare local[number],lpse[number],lpeindex[number]

store space(15) to mstate

@9,5 say "name of state:" get mstate

read

append blank

replace state with mstate

k=1

tlpeindex = 0

spse = 0

do while k <= number

store space(15) to local[k]

lpse[k] = 0

k = k + 1

enddo

k = 1

do while k <= number

@13,5 say "local government:" get local[k] pict "@!"

@15,5 say "primary school enrolment:" get lpse[k] pict "9999999999"

read

k = k + 1

enddo

k = 1

do while k <= number

replace spse with (spse + lpse[k])

k = k + 1

enddo

k = 1

do while k <= number

lpeindex[k] = (lpse[k]/spse)*100

replace tlpeindex with (tlpeindex + lpeindex[k])

k = k + 1

enddo

clear

k= 1

?space(20), "PRIMARY SCHOOL ENROLNMENT FOR", state

?space(5) + "S/NO LOCAL GOVERN. PRIMARY ENROLMENT INDEX "

?space(5), replicate ('=',60)

do while k <= number

?k, '|', local[k], ' | ', str(lpse[k],10), ' | ', str(lpeindex[k],6,2)

k = k + 1

enddo

?space(5), replicate ('=',60)

? space(41) + "TOTAL=>" ,ROUND(tlpeindex,1)

set alternate off

close data

return

*LOCAL GOVERNMENT HOSPITAL BEDS

```
procedure Lshb
set alternate to "A:lhs.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use localhb
declare local[number], lhb[number], lhbindex[number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

k=1
tlhbindex = 0
shb = 0
do while k <= number
  store space(15) to local[k]
  lhb[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  @15,5 say "local government hospital bed:" get lhb[k] pict "9999999999"
  read
  k = k + 1
enddo
k = 1
do while k <= number
  replace shb with (shb + lhb[k])
  k = k + 1
enddo
k = 1
do while k <= number
  lhbindex[k] = (lhb[k] / shb) * 100
  replace tlhbindex with (tlhbindex + lhbindex[k])
  k = k + 1
enddo
clear
k= 1
?space(20), "LOCAL GOVERNMENT HOSPITAL BEDS FOR", state
?space(5) + "S/NO LOCAL GOVERN. HOSPITAL BED INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|', local[k], ' | ', str(lhb[k],10), ' | ', str(lhbindex[k],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tlhbindex,1)
set alternate off
```

```
close data
return
```

```
procedure lrev
set alternate to "A:ligr.out"
set alternate on
set talk off
do header
clear
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use localr
declare local[number], year[number,2], rev[number,2], diff[number], lrindex[numb
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

k=1
tlrindex = 0
tdiff = 0
tperdiff = 0
do while k <= number
  store space(15) to local[k]
  year[k,1]= 0
  year[k,2] = 0
  rev[k,1] = 0
  rev[k,2] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  @15,5 say "first year:" get year[k,1] pict "9999"
  @17,5 say "second year:" get year[k,2] pict "9999"
  @19,5 say "first revenue:" get rev[k,1] pict "9999999999"
  @21,5 say "second revenue:" get rev[k,2] pict "9999999999"

  read
  k = k + 1
enddo
k = 1
do while k <= number
  if rev[k,1] > rev[k,2]
    diff[k]= 0
  else
    diff[k] = rev[k,2] - rev[k,1]
  endif
  replace tdiff with (tdiff + diff[k])
  k = k +1
enddo
k = 1
do while k <= number
  perdiff[k] = (diff[k]/rev[k,1])*100
  replace tperdiff with (tperdiff + perdiff[k])
```

```

k = k + 1
enddo
clear
k = 1
do while k <= number
  lrindex[k] = (perdiff[k]/tperdiff)*100
  replace tperdiff with (tperdiff + perdiff[k])
  replace tlrindex with (tlrindex + lrindex[k])
  k = k + 1
enddo
clear

```

```

k= 1
?space(20)," INTERNAL GENERATED REVENUE FOR", state
?space(5) + "S/NO | LOCAL GOVERN. | 1_REV | 2_REV |"
?space(5), replicate ('=',100)
do while k <= number
  ?k, '|', local[k], '|', str(rev[K,1],10), ' | ', str(rev[K,2],10), ' | ', str(di
  k = k + 1
enddo
?space(5), replicate ('=',100)
? space(100) + "TOTAL=>" ,ROUND(tlrindex,1)
set alternate off
close data
return

```

*LOCAL GOVERNMENT TERRIAN

```

procedure lterrian
set alternate to "A:lt.out"
set alternate on
set talk off
clear
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use localt
declare local [number], rs [number], rl [number], rh [number].
declare ws [number], wl [number], wh [number], tw [number], i [number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate
k=1
gt = 0
gti=0
do while k <= number
  store space(15) to local[k]
  rs[k] = 0
  rl[k] = 0
  rh[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number

```

```

@13,5 say "local government:" get local[k] pict "@" .
@15,5 say "areal proportion of wetlands:" get rs[k] pict "9999999999"
@17,5 say "areal proportion of plains:" get rl[k] pict "9999999999"
@21,5 say "areal proportion of highland:" get rh[k] pict "9999999999"
read
k = k + 1
enddo
k = 1
do while k <= number
  ws[k] = rs[k] * 2.0
  wl[k] = rl[k] * 1.0
  wh[k] = rh[k] * 1.5
  tw[k] = ws[k] + wl[k] + wh[k]
  replace gt with (gt + tw[k] )
  k = k + 1
enddo
k = 1
do while k <= number
  i[k] = (tw[k]/gt)*100
  replace gti with ( gti + i[k] )
  k = k + 1
enddo
clear
k= 1
?space(20),"LOCAL GOVERNMENT TERRIAN FOR", state
?space(5) + "S/NO | LOCAL GOVERN. | ALL TERRIAN | INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',local[k],' | ',str(tw[K],10),' | ',str(i[K],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(gti,1)
set alternate off
close data
return

```

*LOCAL GOVERNMENT EQUALITY

procedure lequal

set alternate to "A:lq.out"

set alternate on

set talk off

clear

do header

use number

store 0 to mnumber

@11,5 say "number of local government:" get mnumber pict "999"

read

number = mnumber

use localq

declare local[number]

store space(15) to mstate

@9,5 say "name of state:" get mstate

read

append blank

replace state with mstate

k=1

tei = 0

```

do while k <= number
  store space(15) to local[k]

  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  read
  k = k + 1
enddo

k = 1
do while k <= number

  replace lindex with (1.0/number)*100
  replace tei with (tei + lindex)
  k = k + 1
enddo
clear
k= 1
?space(20),"LOCAL GOVERNMENT EQUALITY FOR", state
?space(5) + "S/NO | LOCAL GOVERN. | INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',local[k],' | ',str(lindex,6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tei,1)
set alternate off
close data
return

```

*LOCAL GOVERNMENT RAINFALL PROPORTION

```

procedure lrain
set alternate to "A:lr.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local governemt:" get mnumber pict "999"
read
number = mnumber
use localra
declare local [number], ar[number], rp[number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

```

```

k=1
tar = 0
trp = 0
do while k <= number
  store space(15) to local[k]
  ar[k] = 0

```

```

k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  @15,5 say "average rainfall distribution (3 years):" get ar[k] pict "9999999"
  read
  k = k + 1
enddo
k = 1

do while k <= number
  replace tar with (tar + ar[k])
  k = k + 1
enddo
k = 1
do while k <= number
  rp[k] = (ar[k]/tar)*100
  replace trp with (trp +rp[k])
  k = k + 1
enddo
clear
k= 1
?space(20), "LOCAL GOVERNMENT RAINFALL PROPORTION FOR", state
?space(5) + "S/NO | LOCAL GOVERN. | AVERAGE | PROP.RATIO"
?space(5), replicate ('=',60)
do while k <= number
  ?k, '|', local[k], ' | ', str(ar[k],10), ' | ', str(rp[k],6,2)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(trp,1)
set alternate off
close data
return

```

*INVERSE SECONDARY ENROLMENT (LOCAL GOVERNMENT)

```

procedure lienrolment
set alternate to "A:ilse.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use localis
declare local[number], lse[number], inverse[number], ldindex[number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate

k=1
tinverse = 0

```

```

tldindex = 0
do while k <= number
  store space(15) to local[k]
  lse[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@!"
  @15,5 say "secondary school enrolment:" get lse[k] pict "9999999999"
  read
  k = k + 1
enddo

k = 1
do while k <= number

  inverse[k] = (1/lse[k])

  k = k + 1
enddo

k = 1
do while k <= number
  replace tinverse with (tinverse + inverse[k])
  k = k + 1
enddo

k = 1
do while k <= number
  ldindex[k] = (inverse[k]/tinverse)*100
  k = k + 1
enddo

k = 1
do while k<= number
  replace tldindex with (tldindex + ldindex[k])
  k = k + 1
enddo

clear
k= 1
?space(20),"LOCAL GOVERNMENT SECONDARY SC.ENROL.(INVERSE) FOR", state
?space(5) + "S/NO | LOCAL GOVERN. | SEC.ENROL.(INVERSE) | INDEX "
?space(5), replicate ('=',60)
do while k <= number
  ?k,'|',local[k],' | ',str(inverse[k],11,4),' | ',str(ldindex[k],11,4)
  k = k + 1
enddo
?space(5), replicate ('=',60)
? space(41) + "TOTAL=>" ,ROUND(tldindex,1)
set alternate off
close data
return

```

```

*INVERSE RAINFALL PROPORTION(LOCAL GOVERNMENT)
procedure lirain

```

```

set alternate to "A:ilr.out"
set alternate on
set talk off
clear
do header
use number
store 0 to mnumber
@11,5 say "number of local government:" get mnumber pict "999"
read
number = mnumber
use localir
declare local[number], arv[number], iinverse[number], ldiindex[number]
store space(15) to mstate
@9,5 say "name of state:" get mstate
read
append blank
replace state with mstate
k=1
tiinverse = 0
tldiindex = 0
do while k <= number
  store space(15) to local[k]
  arv[k] = 0
  k = k + 1
enddo
k = 1
do while k <= number
  @13,5 say "local government:" get local[k] pict "@"
  @15,5 say "average rainfall over(3 years):" get arv[k] pict "9999999999"
  read
  k = k + 1
enddo

k = 1
do while k <= number

  iinverse[k] = (1/arv[k])

  k = k + 1
enddo

k = 1
do while k <= number
  replace tiinverse with (tiinverse + iinverse[k])
  k = k + 1
enddo

k = 1
do while k <= number
  ldiindex[k] = (iinverse[k]/tiinverse)*100
  k = k + 1
enddo

k = 1
do while k <= number
  replace tldiindex with (tldiindex + ldiindex[k] )
  k = k + 1
enddo
clear
k= 1
?space(20), "LOCAL GOVERNMENT RAINFALL PROP.(INVERSE) FOR", state
?space(5) + "S/NO | LOCAL GOVERN. | AVEG.RAINFALL(INVERSE) | INDEX "

```


LOCALE.DBF
(LOCAL GOVERNMENT SECONDARY SCHOOL ENROLMENT . DBF

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	ISE	Numeric	10		N
4	LEINDEX	Numeric	6	2	N
5	TLEINDEX	Numeric	6	2	N
6	SSSE	Numeric	10		N

LOCALR. DBF
(INTERNALLY GENERATED REVENUE .DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	YEAR	Numeric	4		N
4	REV	Numeric	10		N
5	DIFF	Numeric	10		N

LOCALIR.DBF

(LOCAL GOVERNMENT RAINFALL PROPORTION(INVERSE) . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	ARV	Numeric	10		N
4	IINVERSE	Numeric	11	4	N
5	TIINVERSE	Numeric	11	4	N
6	LDIINDEX	Numeric	11	4	N
7	TLDIINDEX	Numeric	11	4	N

LOCALIS . DBF
(LOCAL GOVERNMENT SECONDARY SCHOOL ENROLMENT (INVERSE) . DBF)

Field	Field Name	Type	WidthDec	Index
1	STATE	Character	15	N
2	LOCAL	Character	15 15	N
3	ISE	Numeric	10 10	N
4	INVERSE	Numeric	11 4	N
5	LDINDEX	Numeric	11 4	N
6	TLDINDEX	Numeric	11 4	N
7	TINVERSE	Numeric	11 4	N

LOCALRA . DBF

(LOCAL GOVERNMENT RAINFALL PROPORTION (DIRECT) . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	AR	Numeric	10		N
4	RP	Numeric	6	2	N
5	TAR	Numeric	10		N
6	TRP	Numeric	6	2	N

LOCALQ . DBF
(LOCAL GOVERNMENT EQUALITY . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	TEI	Numeric	6	2	N
4	LINDEX	Numeric	6	2	N

LOCALT . DBF
(LOCAL GOVERNMENT TERRAIN . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	RS	Numeric	10		N
4	RL	Numeric	10		N
5	RH	Numeric	10		N
6	WS	Numeric	10		N
7	WL	Numeric	10		N
8	WH	Numeric	10		N
9	TW	Numeric	10		N
10	I	Numeric	6	2	N
11	GT	Numeric	10		N
12	GTI	Numeric	6	2	N

LOCALHB . DBF
(LOCAL GOVERNMENT HOSPITAL BED . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	LHB	Numeric	10		N
4	LHBINDEX	Numeric	6	2	N
5	TLHBINDEX	Numeric	6	2	N
6	SHB	Numeric	10		N

LOCALPE . DBF

(LOCAL GOVERNMENT PRIMARY SCHOOL ENROLMENT .DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	LPSE	Numeric	10		N
4	LPEINDEX	Numeric	6	2	N
5	SPSE	Numeric	10		N
6	TLPEINDEX	Numeric	6	2	N

LOCAL .DBF
(LOCAL GOVERNMENT LAND MASS . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	LLAND	Numeric	10		N
4	LIINDEX	Numeric	6	2	N
5	TLIINDEX	Numeric	6	2	N
6	SLAND	Numeric	10		N

LOCALS . DBF
(LOCAL GOVERNMENT POPULATION . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	LOCAL	Character	15		N
3	LPOP	Numeric	10		N
4	LINDEX	Numeric	6	2	N
5	SPOP	Numeric	10		N
6	TLINDEX	Numeric	6	2	N

STATER . DBF
(INTERNAL GENERATED REVENUE . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	YEAR	Numeric	4		N
3	REV	Numeric	10		N
4	DIFF	Numeric	10		N
5	SRINDEX	Numeric	6	2	N
6	PERDIFF	Numeric	6	2	N
7	TDIFF	Numeric	6	2	N
8	TPERDIFF	Numeric	6	2	N
9	TSRINDEX	Numeric	6	2	N

STATEHB . DBF
(HOSPITAL BEDS . DBF)
(STATE)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	SHB	Numeric	10		N
3	CHB	Numeric	10		N
4	SHBINDEX	Numeric	6	2	N
5	TSHBINDEX	Numeric	6	2	N

PRIMAARY SCHOL ENRLMENT . DBF
(STATEPE . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	PSE	Numeric	10		N
3	CPSE	Numeric	10		N
4	SPEINDEX	Numeric	6	2	N
5	TSPEINDEX	Numeric	6	2	N

STATE . DBF
SECONDARY SCHOLL ENROLMENT . DBF
(DIRECT)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	SSE	Numeric	10		N
3	CSSE	Numeric	10		N
4	SEINDEX	Numeric	6	2	N
5	TSEINDEX	Numeric	6	2	N

STATE . DBF

LAND MASS (STATE) .DBF

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	SLAND	Numeric	10		N
3	CLAND	Numeric	10		N
4	SIINDEX	Numeric	6	2	N
5	TSIINDEX	Numeric	6	2	N

STATES . DBF
(POPULATION (STATE) . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	SPOP	Numeric	10		N
3	CPOP	Numeric	10		N
4	SINDEX	Numeric	6	2	N
5	TSINDEX	Numeric	6	2	N

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	SPOP	Numeric	10		N
3	CPOP	Numeric	10		N
4	SINDEX	Numeric	6	2	N
5	TSINDEX	Numeric	6	2	N

STATERA . DBF
(RAINFALL PROPORTION (DIRECT) . DBF)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	AR	Numeric	10		
3	RP	Numeric	6	2	N
4	TAR	Numeric	6	2	N
5	TRP	Numeric	6	2	N

STATEIS . DBF
(SECONDARY SCHOOL ENROLMENT)
(INVERSE)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15		N
2	SSE	Numeric	10		N
3	INVERSE	Numeric	11	4	N
4	TINVERSE	Numeric	11	4	N
5	SDINDEX	Numeric	11	4	N
6	TSDINDEX	Numeric	11	4	N

STRUCTURE OF (EQUALITY) . DBF
(STATE)

Field	Field Name	Type	Width	Dec	Index
1	STATE	Character	15	2	N
2	INDEX	Character	6	2	N
3	TEI	Character	6		N

TERRAIN . DBF
(STATE)

Field	Field Name	Type	Width	Dec	Index
1	RS	Numeric	10		N
2	RL	Numeric	10		N
3	RH	Numeric	10		N
4	WS	Numeric	10		N
5	WL	Numeric	10		N
6	WH	Numeric	10		N