DESIGN OF A COMPUTERIZED PROCEDURE FOR COMMERCIAL LOAN ANALYSIS

By

DAVID L. LAMB

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1974

Submitted to the Graduate Faculty of the
Department of Administrative Sciences of
the College of Business Administration
of the Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF BUSINESS ADMINISTRATION
December, 1975

DESIGN OF A COMPUTERIZED PROCEDURE FOR COMMERCIAL LOAN ANALYSIS

Report Approved:

Winfield P. Beth
Report Adviser

Head, Department of Administrative Sciences

Name: David L. Lamb Date of Degree: January, 1976

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: DESIGN OF COMPUTERIZED PROCEDURE FOR COMMERCIAL LOAN ANALYSIS

Pages in Study:99 Candidate for Degree of Master of Business Administration

Major Field: Finance

Purpose of Study: The purpose of this study is a discussion of existing techniques of commercial loan analysis and to develop a computer model to assist loan officers in the evaluation of credit decisions. The intent of the model is to present an orderly and compact arrangement of the financial data used to analyze loan requests. The model uses a IBM/CPS system. This system provides the model with real time conversational inputs and outputs, combined with the speed and power of computers.

Findings and Conclusions: It was found that there is not one correct method or technique for evaluation of loan requests, but a computer model can be developed that can provide assistance to the loan officer. The model produces financial outputs to help analyze the creditworthiness of a borrower. Although the model does produce outputs that help to evaluate the repayment capability of the firm, the model does not and should not concern itself with the actual credit decision. The procedures developed provides the loan officer with the flexibility that they need to make accurate credit decisions. By computerizing financial data needed in loan decisions, the loan officer is able to spend more time on those activities at which only men are adept.

ADVISER'S APPROVAL _____

PREFACE

This paper is concerned with the development of a computer model to assist banks in commerical loan analysis. Reasons for developing and using computerized bank models are presented along with a review of existing loan techniques and computer models.

I wish to express my appreciation to Dr. Winfield P. Betty for his assistance in the development of this paper.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Commerical loans are among banking's most important services to the business community. They are also among banking's major source of income. Yet, despite the unquestioned importance of the commerical loan function, it is one of the last banking activities to be automated. The reason for this lag appears to be the complexity of designing a comprehensive data processing system for commerical loans and insufficient understanding of the potential opportunities for improving the information available.

Where automated systems have been developed, designers have concentrated on the traditional loan accounting and servicing activities, rather than on systems to facilitate decision making. While banks have been slow to evolve such systems, the information needs of commercial loan officers and management have increased. The art of lending is rapidly becoming very complex. The analysis of a customer's financial statement, financing needs, repayment abilities and the monitoring of complex loan agreements are more difficult than ever. The continuing high demand for loans and the increasing cost of funds have also emphasized the need for greater analysis of customer loans.

In accordance with the views expressed above, banks are becoming increasingly concerned about the total relationship with their commercial

customers. The commercial loan officer is becoming more and more like an account representative, offering a wide range of financial services and acting as a financial consultant to his customers. But, due to the clerical accounting orientation of traditional systems, much of the information he needs to perform most effectively is not available. The construction of an automated information system is a possible answer to this problem.

While there are certain activities in commercial lending that cannot be automated, an analysis of the typical loan officer's activities will show that a disproportionate amount of his time is spent scanning, collecting, and correlating customer financial data. To a large extent, these processes can be automated to provide the loan officer and the commercial loan administrator with meaningful, readily-usable information. By computerizing this information, the loan officers will be able to spend less time on those clerical tasks that machines can do and more time on those activities at which men are particularly adept.

Purpose

The purpose of this study is to build a computerized commercial credit analysis model and information system for loan officers to use in analyzing loan requests. The model to be built is constructed in a flexible manner so that nonessential parts of the model are brought into analysis only when needed. An implicit assumption is made that the presentation of irrelevant information only complicates analysis; whereas, in more complex lending situations, all relevant data should be potentially available. Under such assumptions, the amount and types of information used are left to the discretion of the user of the programs.

Approach

The model that is developed is applicable to almost all types of companies. It uses common financial statements, ratio analysis and regression analysis of past trends in analyzing the credit-worthiness of loan customers. Furthermore, the information requested by the program and it's output are presented in an orderly sequence to facilitate analysis.

Implementation of the model involves execution of a series of time-sharing computer programs. There are four main reasons for using the time-sharing approach:

1) Banks can avoid the need for expensive computer programs since such systems usually contain a library of programs.

2) Through time-sharing, operational efficiencies are attained.

3) The user has conversational ability with the computer.

4) On-line processing, therefore producing an indepth credit analysis with a minimum of effect and time.

The programs are written in CPS (Conversational Programming System) which is a sub-set of full PL-1. Communication with the computer is through typewriter-like terminals and acoustic couplers. This system gives the user the impression that the computer is responging to him alone because the terminals provide nearly instantaneous processing.

The CPS programs are converted to machine language by an interpretive compiler provided by IBM. The hardware used includes an IBM 2741 typewriter-like terminal connected to a IBM 360-65 computer through an IBM 2702 control unit. This particular combination of hardware and software gives the advantages that are desirable in a computer credit

analysis model such as economy of operation, response speed, and an accurate and orderly method of arranging financial data.

Organization of the Paper

Chapter One has dealt with purpose and approach. In this chapter the need for automation of credit analysis procedures was suggested. Also, the approach used in developing a related computer model was outlined. Chapter Two is concerned with a review of the literature related to how loan officers in commercial banks make their credit decisions. Also, in this chapter, some of the existing computerized credit models are presented showing their weaknesses and strengths. Chapter Three describes the design of the model, objectives of the model, what inputs are required, and the different types of financial statements and projections included in the output. The next chapter, Chapter Four, describes how to use the model and some of the major points to be analyzed in each statement to determine the soundness of credit customers. Finally, Chapter Five draws conclusions and summarizes the paper.

CHAPTER II

REVIEW OF EXISTING LOAN ANALYSIS TECHNIQUES AND OTHER RELEVANT LITERATURE

Development of a systematic procedure for the analysis of credit requests is the purpose of this study. Most loans requested by firms are to finance expanded assets and seasonal needs in order to increase the sales of that firm. For these situations, a systematic procedure is needed because of the complexity of analyzing the borrower's credit-worthiness and repayment ability.

In this chapter, financing problems are discussed from the point of view of the banker and various techniques of loan analysis are examined. After this, comes a discussion of existing computer models for loan analysis.

Credit Analysis - Banker's Point of View

Can old commercial loan analysis methods of the past provide adequate data for future analysis? This question was addressed by Dow Ostland (1) when he stated, "We are no longer witnessing temporary changes in the business climate which will disappear when other financing fads come along. Rather, we are seeing fundamental alterations in many of our methods of analyzing loan requests and doing business."

Ostland (1) further states, that there are three main areas of change:

1) lending for longer terms,

2) closer relationship with customer operations,

3) changes in the methods of loan administration.

What do these changes mean to bankers? Longer term loans will be paid back from future operations, not current profits; therefore, more emphasis should be placed on future cash flows and overall soundness of the business, rather than on current operations, Liquidity and current ratios will continue to be important in predicting short-run credit-worthiness and bankers will look closer at adequate capitalization.

Customers will expect bankers to fulfill their self-declared advisory role more completely than in the past. Seven or ten years is a very long time for any banking organization to risk it's money. Alone with the customer's needs for more advice, banks will certainly require a greater role in management because of loan length.

When loan customers come to borrow, they will expect little time to elapse before they receive an answer to their request. This implies that banks will need to develop more sophisticated and efficient loan review procedures.

The risk associated with debt financing is increasing rapidly according to Vandell (3). The environment of the world and the constraints that businesses must operate in are making the commercial loan analysis far more difficult. Many corporations are seeing their cash inflows becoming very unstable, causing many problems in the area of debt management and loan repayment.

As evidenced by the above discussion, the environment that bankers have worked in is changing rapidly, requiring them to develop new procedures and means of loan analysis.

What framework should banks use to evaluate loan requests? According to Houget (4), there are three topics to look at in term loan analysis:

1) analyzing the credit-worthiness of the borrower,
2) structuring the loan,
3) pricing the loan or evaluating the risk associated with a given loan.

He further states that the main component to be analyzed is a prediction of the cash flow in the future and the probability that such predictions are correct. Houget shows how the term "cash flow" has been abused and has become confused with working capital flows. Cash flow is not net profit plus depreciation, but also includes the changes in current assets and liabilities.

A method discussed by John Andealis (5) recommends the basic "Four C" approach of:

1) capacity of potential customers to service his debt,
2) the collateral at his disposal,
3) his overall credit rating,
4) and his character and reputation in the market place.

Another approach by Hunn (6), of people, purpose, payment, protection, and perspective, is referred to as the "Five P" approach. According to Hunn, once the data are gathered, this frame work enables a conclusion to be reached rapidly in each category with people and payment normally carrying the most weight.

In an article by P. S. Cooper (7) in which he discusses the need of loan structuring, the importance of the Four Cs of credit is stressed but he implies that a Five P approach provides an inside look at the borrower and provides greater flexibility. Daniel Kujawa (8) in his article about how bankers approach business loans states:

> "The traditional approach has been to consider and evaluate the Three Cs of credit, namely character, capacity and capital. Over the years, other Cs have been added, such as conditions (economic), collateral and competition (relative strength of the borrower in relation to his competitor). This approach has alliterative appeal, particularly in remembering the key factors. However, in my opinion, the alliteration tends to obliterate more direct and relevant considerations, and doesn't provide for a systematic and well organized approach."

That approach that Kujawa prefers is outlined and discussed by James A. Ensign (9) in an article entitled, "The Elements of Unsecured Business Loans". In this article, Ensign (10) delineates the four questions that must be answered in virtually every request for a business loan:

1) What is the quality (integrity and ability) of management?

2) What is the specific purpose of the loan?

3) What is the planned time and source of repayment?

4) What alternative sources of repayment, if any, are needed?

As evidenced by the above discussion, the approach by Ensign is very similiar to the Hunn approach of evaluating loan requests. Hunn (11), Kujawa (12), Ensign (13), all believe that the Four or Five P approach to loan structuring is the most effective method. The approach of these authors has considerable appeal, and provides the basis for the approach recommended in this paper.

People

People are the borrowers of money, whether they be individuals, or representatives of partnerships or corporate entities. Hunn (14) implies that people should be appraised on two bases: are they responsible, successful businessmen and do they treat their bankers equitable? According to Kujawa (15), the quality of management is perhaps the single most important factor in the success or failure of any business. Kujawa(15) further states that this factor is sometimes overlooked and frequently not given enough attention. The reason that this factor is frequently overlooked is that integrity and management ability, being intangible, are difficult to measure. Integrity, like ability, is a relative quality, and cannot be precisely measured, but can be gauged in an approximatic range. No one is totally honest when they come to a bank to borrow money, but fortunately most people fall somewhere in the favorable end of the scale.

There are many sources that can provide clues to a borrower's integrity. An investigation of the loan seeker's background through normal sources of suppliers, competitors and professional relationships will reveal whether they are responsible. Financial statements can also provide clues. For example, if a borrower requests $30,000 to pay income taxes and a review of the current financial statement shows a tax liability of only $18,000, obvious questions should be asked. Under such conditions it is possible that the borrower is not being honest with the loan officer and further questioning might reveal that the money is needed to pay other overdue liabilities. As another example, a small business owner might agree with the bank to curtail his annual bonus in order to build up

capital for growth. A later statement shows that the borrower has kept his promise partially. The balance sheet reveals that the borrower has acquired a new asset, a $5,000 car. He had cut his bonus, but has acquired a new car. From these examples, it would seem that a person's integrity can sometimes be approximated by reviewing what the borrower has borrowed money for in the past and whether or not he has used the money borrowed for the reasons stated.

The question of integrity and equitability by a borrower is demonstrated by his willingness as well as ability to repay past loans, in his reasonable attitude in sharing appropriate credit information, and in the general relationship between the bank and the loan customers.

Financial statements can also help to cast light on the ability of management. An evaluation of operating trends and returns on equity and total assets relative to their competitors are good indexes of success of management. This will not always be a good technique because of difficulty in adjusting statements for different accounting methods, so a closer examination of accounting methodology can sometimes be required.

The ability as well as honesty of borrowers is very important and should be scrutinized very closely for signs, whether they be positive or negative. Kujawa (17) states, "The debtors quality of management is the most important factor in lending." If management has been unsuccessful in the past, there is a good indication that this trend will continue in the future.

In summary, as the authors above indicate, the integrity and ability of borrowers is significant in determining credit-worthiness. The banker

must acquire all the salient information available about the loan request and evaluate the information diligently before making a credit decision.

Purpose

The intended use of a loan must be evaluated on the basis of the need and adequacy of the amount requested to serve that need. This might involve a judgment by the banker as to the overall economic benefits to be derived from expansion of operations, or from modernizing for improvement in efficiency. According to Hunn (18), from a balance sheet standpoint, the purpose of every bank loan can be assigned to one of three catagories:

1) acquire assets,

2) replace other creditors,

3) replace or increase equity.

When assets are acquired, they are acquired for three purposes and these purposes must be identified:

1) For current assets of a seasonal nature, such as inventory fluctuations.

2) For current assets of a non-seasonal nature, such as the need for more work capital as a business grows.

3) For non-current assets, normally fixed assets used for increasing production capacity.

The banker is concerned with the wisdom of the proposed purpose of the loan and the effects of the loan on the financial conditions of the business. A loan to buy increased inventory in connection with a large manufacturing order is a normal purpose, but a loan to buy securities may be imprudent for both the bank and the borrower.

Many financial authors discuss the compatability of the types of funds used in relation to the nature of assets financed. Johnson (19) used the term "suitability" and showed some of the problems encountered with unsuitable financing. Johnson, reiterates (20) the general rule of financing permanent assets with permanent funds, and sets as an objective for the financial managers, the financing of temporary current assets with flexible short term debt.

Many industrial firms have a long term rising sales trend coupled with cyclical and seasonal variations in sales. Under such conditions, total permanent assets rise steadily in the form of current assets and fixed assets. Weston and Brigham (21) stress these "financing patterns" in their discussion of working capital needs of a growing firm. They state that such increases in permanent assets are normally financed by long-term debt and equity. On the other hand, Weston and Brigham state that temporary increases in assets should be covered by short-term liabilities. They show that if these basic rules are disregarded, a profitable firm may become unable to meet cash obligations or suffer from periodic excesses of cash.

Full knowledge of the exact purpose of the loan proceeds will greatly assist in determining the correct answer to repayment. In many cases, businesses finance long-term asset expansion with short-term money. This causes many problems in the cash flow needs of a business. A banker must not only evaluate the purpose of the current loan request, but also the property of past financing.

Payment

The real test of a lending officer is his ability to establish a repayment schedule. The repayment schedule is usually derived from analysis of cash flows related to the purpose of the loan, so that the cash flows generated from the loan are sufficient to meet cash repayment without crippling the liquidity of the firm receiving the loan. Repayment of a loan should be based on cash generated from borrower's own resources in the ordinary course of business with no reliance for payment placed on sale of colleteral or call upon endorsers.

In essence, the lending officer must analyze the sources and timing of repayment and be convinced that the probability of repayment is high. The difficulty in analyzing repayment is that the lender is dealing in the future, and is armed only with information from the past. The past, however, will usually suggest the probabilities of the future success of the company.

There are only four possible sources of repayment for any loan (22):

1) conversion of assets to cash,
2) cash resulting from profits earned and retained in the business (new capital internally generated),
3) cash resulting from additional investments by stockholders (new capital from external sources),
4) transfer of debt to another creditor.

Most loans made by banks are in the form of cash-flow loans. That is, repayment should be based on future profits or seasonal working capital turnover. The size of these cash flows can frequently be estimated from pro-forma statements.

In summary, identifying the revenue-generating capacity of the borrower's assets within a seasonal, industrial and economic context is the responsibility of the banker. Also, understanding the underlying determinants of the borrower's cash flow behavior is essential in establishing repayment ability.

Protection

Protection implies the taking of appropriate kinds of collateral, in negotiable form when possible, as a support for credit risks. There are two types of protection——internal and external. With internal protection the lender looks exclusively to the borrower. In the case of external protection, a third party adds it's credit responsibility to that of the borrower. Internal protection implies the pledging of securities, inventories or other assets of the business requesting the loan. External protection, which is only a secondary thought in this discussion, usually takes the form of guarantees, endorsements or certain types of repurchase agreements.

Collateral or "protection" alone is rarely a sufficient basis for making a loan. Collateral is taken to offset some deficiencies in one or more areas of a firm. A good unsecured loan in most cases is many times better than a secured loan. According to Kujawa (23), the best protection a loan officer can have is to deal with an honest and competent businessman. Namely:

1) one who borrows for valid purposes,
2) one who clearly knows when and how he can repay the loan,
3) one who has alternative sources of repayment available,
4) and one whose business has a solid financial condition.

Perspective

Perspective is the last of the Five Ps, and one of the most important. The loan officer must review the four preceeding Ps and determine if the risk and reward of a loan request fits into the total framework of the bank. The loan officer, along with analyzing the risk of principal loss, must also determine the possible opportunity loss of being "locked in" with a borrower. The bank must weigh the loan request against it's current loan portfolio to make sure an appropriate balance is maintained between different types of loans. Only loans that are consistent with the bank's own loan policy should be accepted.

REVIEW OF EXISTING COMPUTER MODELS FOR LOAN ANALYSIS

The allocation of credit is thought by many to be the primary function of banks. All bankers must be concerned with various aspects of credit problems. How can the banker distinguish good credit risks from poor credit risks? What general procedures should the banker follow when dealing with prospective commercial loan customers? These general questions and means for identifying the answers to these questions have been built into computer programs with varying degrees of success for the last ten to fifteen years.

In this section of the paper an overview of relevant computer models used in commercial loan analysis, and loan selections techniques are discussed. The existing computer models can be divided into three general categories. These categories are:

1) linear programming model for loan selection,

2) credit scoring models,

3) computerized credit analysis models.

Type one and two are selection-type models, not involving personal analysis to the same extent as number three. The following discussion considers all three types of models showing weaknesses, strengths, and designs of models that have been developed in each category.

Linear Programming Models

Linear programming is a technique that maximizes or minimizes a linear objective function subject to a set of linear constraints. Problems solved with linear programming belong to a class of allocation problems that arise when a number of tasks have to be performed subject to a variety of constraints on resources.

The uses of L-P are aimed at the total loan portfolio rather than at individual analysis to determine credit-worthiness. L-P models were among the first models to be computerized by banks. Because portfolio selection is not the scope of this paper, a very general overview will be given concerning these types of models.

There are many different models developed in the last few years using L-P to evaluate loan selection. Although the models are somewhat different they all rely on the same basis of allocating resources among competing uses. Because of this similiarity between models, the discussion will center on a model developed by Robert Waterman and Robert Gee (24). Anyone wishing to learn more about the uses of L-P for bank models can be referred to Orgler (25), Eilon and Fowles (26), Morris (27), and Williams (28).

Commerical banks are confronted with several objectives. These

objectives are often in conflict with each other. For instance, bank management would like to maximize revenues, but realizes that safety and service to the community are also important. To complicate matters further, the bank must operate under a large number of legal and policy restrictions. As a result of these constraints, the problem of how best to invest a bank's funds is complex and subject to a great deal of judgment.

According to Waterman and Gee (29), "Linear programming is a systematic way of finding the best course of action when many variables and many conditions must be taken into consideration." Stated another way, it is an approach to satisfy an objective subject to many restrictions. The objective of most banks is profit. The other objectives of the bank such as safety and community service can be viewed as part of the network of restrictions. The remaining restrictions are composed of general economic conditions, legal constraints, and bank policies.

The L-P models developed so far are used mainly in helping banks to allocate their loan portfolio among different types of loan categories (unsecured loans, secured loans, real estate loans, etc.).

Since L-P models are not aimed at individual loan requests, but, are general in nature dealing with the problems of allocation of funds, a further discussion of these types of models is not within the intended scope of this paper. It is worth noting, however, that L-P has contributed heavily in developing systematic models giving banks insight into their everyday operations and in evaluating these operations.

Credit Scoring

Numerical credit scoring is a concept that has been well-known for years to those responsible for making installment loans to retail customers. Under this system, points are assigned depending on the answers given to each item in the loan application. The total points for an application are then compared to some critical level, and the loan is made or rejected according to the criterion of whether the score of the application is above or below some preassigned "cut off" point. Whether or not numerical credit scoring is worthwhile depends on the relevance of the criteria, the reliability of the data, and the weights given to the various factors.

An article by Myers and Forgy (30) describes the results of a study in which statistical techniques were employed to determine the relative weights which should be assigned to various questions on an application form. By assigning different weights through computer analysis of different questions, the validity of retail credit scoring was increased substantially.

For several reasons, it is difficult to apply the credit scoring method to the evaluation of commercial loans. Hamner and Orgler (31) discuss these problems. According to these writers, the number of commercial borrowers is normally quite small compared to customers for consumer credit, thus, there is a problem in obtaining a sufficient number of observations for a statistically significant study. Second, there are substantial variations among commercial loans with respect to their size, terms, collateral, and payment procedures, all of which are relatively uniform in the case of comsumer loans. Finally, there is a lack of reliable up-to-date financial data on commercial borrowers, particularly small

business borrowers.

Because of problems similiar to those just mentioned, it is very difficult to develop a general credit scoring model for commercial loan applications. There are however, several interesting models that have been developed recently which appear to be of some value. The discussion below is mainly based on a model now installed at the Indiana National Bank (32). There are many other models. For example, Ewert (33) developed a model for screening trade credit applications by small firms. The model combines information on the owner with data from Dunn and Bradstreet reports. Other studies involving the derivation of credit-scoring models for business loans were developed by Edmister (34) (35) and Abate (36). Another interesting model was developed by Cohen, Gilmore, and Singer (37), that simulates the commercial lending decisions of bank loan officers. In fact, during the past thirty years, credit scoring models have been developed and implemented in a variety of credit granting institutions with mixed results. According to Heffehouse and Wentworth (38), most models appear to fail because of a general lack of creditability or because of their use as a "black box" that ignores human judgment. They believe, however, that credit scoring can be installed successfully.

The Credit Analysis Model (CAM) currently being implemented at Indiana National Bank, utilizes a two-stage decision process. In stage one the banker is able to assign points to the application in accordance with statistically determined weights. Some applications are accepted or rejected automatically, depending on the risk (high or low scores) associated with the loan. Those requiring additional consideration are forwarded to an experienced credit analyst. The analyst then supplies

his judgment in reaching a conclusion. If after the two-stage process, the analyst is unable to reach a conclusion, he may request additional information in the form of credit reports. Loans can be accepted or rejected at three different times during the evaluation process:

1) after a statistical score is applied,
2) after consideration of the secondary variables,
3) after examination of credit reports.

As with any computer model, the value of the model is dependent on the quality and quantity of the input data. In the case of credit scoring models, there are many ways in which the validity or the consistency of the model can be undermined. Most of the problems with credit scoring models were discussed at the start, but the main problem is the inability to define the proper weights to assign each category. The validity of these weights can only be measured from reviewing the loans made using this technique.

The Indiana Bank, understanding the strengths and weaknesses of credit scoring, has developed a computer model which scores loans and then passes the evaluation on to a loan analyst who is able to justify accepting or rejecting the loan. By using personal judgment and the computer's ability to analyze large amounts of data, the loan officer is able to make better credit decisions.

Computerized Credit Analysis Models

These types of models use the computer to make calculations, project trends, and perform repetitious ratio analysis. The actual credit decision is left to the discretion of the analyst. The computer is used

only as a means to facilitate the formulation and presentation of data.

The model discussed by Sangster and Raguso (39) uses a timesharing program which does not have real-time response capability. The model is also limited in the amount of information outputs. They have found that the following reports on a six period basis can be produced for a cost of approximately $25.00 including analyst time:

1) balance sheet and income statements,

2) common size statements,

3) sources and uses of working capital,

4) ratio analysis.

All the above reports involve an automatic currency conversion to U.S. dollars. They estimate that the same reports would take up to eight hours of analyst time to do by hand.

Burd and Blades (40) claim that computers can become a valuable partner of the bank lending officer by providing quick, comprehensive evaluations of borrowers. Burd and Blades have developed a very complete model providing the banker with:

1) balance sheet,

2) income statement,

3) income statement (percent of total sales),

4) ratio analyst,

5) sources and uses statement,

6) schedule showing the interest expense and principal repayment.

Although these statements are comprehensive, the analysis is done by a separate department other than the commercial loan department and requires 24 hours for turn-around.

These authors believe that such models provide flexibility that is needed over manual procedures. Also, it is believed that these models increase the analyst's capabilities due to the number of quantitative factors which can be introduced into credit decisions.

In summary, the literature review has suggested the availability of a mixed variety of information on how to analyze the commercial loan requests. Although there are many articles that relate to bank loan analysis, there is still ample space in the literature for further studies. The areas of financial theories relating to banking institutions appears to lag the theories and models relating to other areas of finance.

In developing the model for this study, many of the concepts and techniques presented in the above review were utilized. The resulting model is intended to systematically incorporate a number of these techniques into a comprehensive analytical loan review procedure.

CHAPTER III

DESIGN OF THE MODEL

The orderly presentation of financial data in making credit decisions is the overall object of this model. The computer outputs provide information showing past performance figures and projections of future cash flows——hopefully projecting a reliable picture of the ability of the borrower to repay principle and interest. In order to produce this information, the model receives a large number of purposeful inputs. The inputs consist of both historical and projected financial data. Once this data is entered, it is manipulated and transformed to meet the specific needs of the user. The computer model is composed of thirteen logic modules which are called into the model when needed. The input data is limited in that only basic or aggregate accounts are built into the programs and the borrower's financial data must be redefined to fit the accounts. The accounts that are created are general in nature to insure that all the necessary balance sheet and income statement data can be entered. The output of the model is constrained somewhat, because of the use of generalized inputs. Generalization of the inputs and outputs are essential in order to utilize the Conversational Programming System (CPS). These accounts are defined in a way so that their aggregate form will have little effect on the validity of the model.

Inputs to the Model

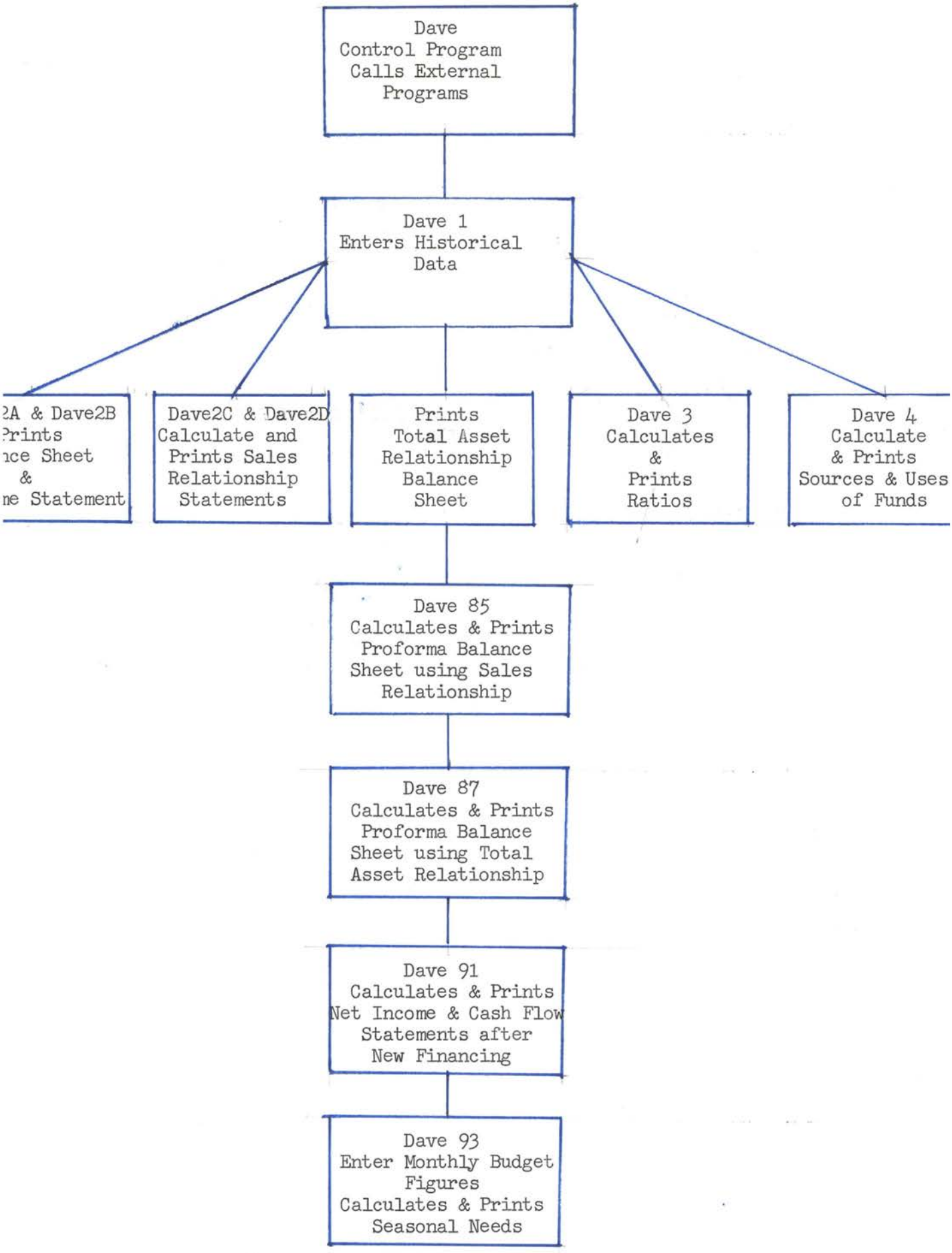
The computer model requires a large number of inputs consisting of:

1) annual balance sheet up to five years,
2) annual income statement up to five years,
3) monthly projected sales for the coming year,
4) monthly projected operating expenses for the coming year,
5) proposed loan amounts and terms, and
6) certain miscellaneous data consisting mainly of yes and no responses.

If organized, this information requires a maximum of fifteen minutes to enter. Once the data is entered, it is stored on temporary disk files for use by the different logic modules.

Components of the Model

The control module calls the appropriate modules in a logical sequence following the instructions of the user. Because of certain technical problems with the O.S.U. computer system, four of the programs must be loaded by the user. These logic modules, after being loaded by the user, perform in the same manner as the other modules. After each function is completed, the control model is in command except for the four modules which must be loaded by the user. A flow chart of each logic module and it's basic function is shown in Figure 1.


Dave
Control Program
Calls External
Programs
Dave 1
Enters Historical
Data
2A & Dave2B
Prints
ıce Sheet
&
ne Statement
Dave2C & Dave2D
Calculate and
Prints Sales
Relationship
Statements
Prints
Total Asset
Relationship
Balance
Sheet
Dave 3
Calculates
&
Prints
Ratios
Dave 4
Calculate
& Prints
Sources & Uses
of Funds
Dave 85
Calculates & Prints
Proforma Balance
Sheet using Sales
Relationship
Dave 87
Calculates & Prints
Proforma Balance
Sheet using Total
Asset Relationship
Dave 91
Calculates & Prints
Net Income & Cash Flow
Statements after
New Financing
Dave 93
Enter Monthly Budget
Figures
Calculates & Prints
Seasonal Needs

Dave

This program contains very little programming logic, but primarily is a series of call statements. Each call statement causes a specific external procedure or logic module to be loaded and executed by the CPU.

Dave is the program that the user loads and executes in CPS. At any point in time, Dave and one of the external procedures are resident in the 360-65 CPS memory. Dave takes one "page" and each logic module is two, three, or four "pages" in size. In the O.S.U. CPS system, only four pages of programs can be in CPU memory by a single terminal user.

This restriction imposed by the O.S.U. computer system causes some inconvenience to the user but, does not substantially effect the model. There are four programs, (Dave 85, Dave 87, Dave 91, and Dave 93), that must be loaded and executed by the user. The control module gives ample instruction for the steps that the user must perform. The files that are created in earlier modules are accessible for use in the four programs.

Dave 1

Dave 1 is the first logic module. This external procedure asks the user to enter the historical balance sheet and income statements for up to five years. A number of elementary balance sheet and income statement calculations that were derived from this data is retained for other logic modules by writing a temporary disk file.

Dave 2A and Dave 2B

Dave 2A uses the temporary files to print comparative balance sheets

for each year of history. Dave 2B prints the comparative income statements and cash flow figures for the historical data entered. The intent here is to provide the user with a compact and orderly record of the historical financial data which the user has furnished the model.

Dave 2C and Dave 2D

Dave 2C and Dave 2D use the temporary files to evaluate and print comparative balance sheets and income statements showing the relationship between individual accounts and sales. The statement is printed in percentages helping the user to better analyze historical trends of the firm being evaluated.

Dave 25

This program uses the temporary files to provide a comparative balance sheet showing the relationship between individual assets and total assets and between individual liabilities and total net worth and liabilities. The intent, as in the earlier modules, is to provide the user with an orderly picture of historical data and to help the user to determine balance sheet trends.

Dave 3

This module calculates and prints sixteen financial ratios from historical data stored on the temporary files. These ratios are listed below.

Liquidity Ratios

1. Current - Ratio
2. Quick Ratio

3. Inventory to Working Capital

Leverage Ratios

4. Debt to Assets

5. Interest expense to earnings before interest expenses and taxes

Activity Ratios

6. Accounts receivable to daily sales

7. Current liabilities to net worth

8. Fixed asset turnover

9. Fixed asset net worth

10. Total asset turnover

11. Cash velocity

12. Inventory turnover

Profitability Ratios

13. Profit margins

14. Operating expenses to sales

15. Return on assets

16. Net worth return

Dave 4

Historical data is accessed from the temporary disk file in Dave 4. This data is used to calculate sources and uses of funds statements, and to show balance sheet changes from one year to the next. Dave 4, besides showing the individual account changes, also shows the total changes in sources and uses of funds.

Dave 85

This module is able to use the historical data to perform regression

analysis on the past sales to form a trend. If the trend of past sales is not adequate, the user may enter his own projected sales figures. By assuming a constant relationship between sales and individual accounts, the computer is able to project the balance sheets for the next five years.

Dave 87

Again, regression analysis is used on historical data to form a trend. This time a projected balance sheet is created by using the trend of total assets and liabilities plus total net worth. If the regression figure is not feasible the user may enter his own projection for these accounts. A constant relationship is assumed between total assets and individual assets accounts with liabilities, plus net worth and individual liabilities accounts also assuming a similiar relationship.

Dave 91

This logical module reads the temporary disk files to get the historical data to project future income and cash flow. This program, as the two preceding programs, uses regression analysis to project trends in future sales. If these projections are not accurate, the user may enter his own sales projections for the next five years. Dave 91 assumes a constant relation between sales and expenses in calculating and printing the pro-forma income statements. The user is required to enter the amount and terms of the loan request and the program shows the cash flow of the firm after payment of existing debt and the new debt payment.

Dave 93

More input to the model is requested by this module. Specifically, the user must provide monthly cash budgets figures, including monthly sales, operating expenses, and a category to include all other monthly expenses. Beginning cash and minimum cash totals must also be entered. The calculation that follows is a simple cash budget calculation on a monthly basis to show the seasonal needs and the size of credit that is needed to meet these seasonal needs. This information is presented in tabular form.

Outputs of the Model

The outputs are of two types and can be classified as historical or pro-forma. The outputs and inputs are intermingled throughout the model in a conversational manner so that flexibility can be provided to the user.

The historical outputs are provided initially and the pro-forma outputs are based on this earlier financial data. Beginning with historical financial statements, the terminal types out annual comparative balance sheets and income statements for up to five years. Next, the model provides the user with comparative balance sheets and income statements showing relationships between sales and individual accounts and between total assets and liabilities plus total net worths. Financial ratios are next calculated and printed from the historical data into a tabular form. Then, the program prints source and uses statements to show application of funds from one year to the next.

After this historical data is presented to the banker in an orderly

fashion, the banker is able to determine the past operating trends of the business. Beginning with pro-forma balance sheets there is a transition between historical and projected data. The next two outputs of the model present five years of pro-forma balance sheets using two different means to project the account totals. The next module provides the user with five years of pro-forma income statements, assuming a sales relationship. The output shows the amount of cash that will be available to meet dividends and future expansion after the new debt payments are paid.

Finally, the computer provides the user with a monthly cash budget showing the surplus or the amount of funds that need to be borrowed to meet seasonal needs.

Summary

The model discussed in the preceding paragraphs involves execution of a series of programs in a conversational mode. The historical inputs are entered first and subsequent printed outputs are arranged in a systematic manner for easy interpretation by the banker. The next chapter describes a sample execution and some of the main factors in the output useful in analyzing a loan request.

CHAPTER IV

SAMPLE EXECUTION OF THE MODEL

The model presented is not intended to replace the loan officer, but to assist in credit decisions. By using this model, the loan officer gains the speed and power of the computer to quickly perform a great many calculations. The results of these calculations are presented in an orderly fashion giving the loan officer information upon which to judge customer credit-worthiness.

This chapter is concerned with showing the user how to use the IBM/CPS system, and how the model can be executed. A credit decision case is presented and the model is executed showing the ability of the computer to help the loan officer make credit decisions. There is a brief discussion of the output of the model, pointing out some of the areas to be analyzed in making credit decisions.

Use of IBM/CPS

Implementation of the model requires access to the proper combination of computer hardware and software. If not otherwise available, such facilities can be leased for a reasonable cost from a computer service organization.

The procedure for using the O.S.U. IBM/CPS is as follows:

1) Turn the IBM 2741 on.

2) Press the "talk clear" button on telephone.

3) Dial the extension telephone number of the computer (7641).

4) When you hear a high screeching tone, indicating the computer has answered, press "Data" button on the telephone.

5) Hit the "Return" key on the terminal.

6) "Login" on the terminal.

7) Give the "Password".

After the user has logged in, they are ready to execute the model. By typing the command "load (Dave)" the user loads the control module. Dave is executed by typing in the command "Xeq". At this command, the control module (Dave) calls the first external procedure and from this point on (except for the technical problems discussed earlier), the procedure is directed by Dave according to the responses of the user.

Case

Limber Lumber Company

The Limber Lumber Company is a wholesaler of hardwood lumber. The lumber company has had steady growth of sales over the past five years, and they predict that sales will continue to increase at about the same rate as in the past. In order to take advantage of the strong market for lumber, Limber Lumber has requested a $100,000 term loan for expansion of facilities and increases in working capital. The loan is to be amortized over a five-year period at a interest rate of thirteen percent.

Sample Execution

The analysis of financial statements consists of a study of relationships and trends to determine whether or not the financial position and results of operations of the company are satisfactory or unsatisfactory.

The analytical methods and techniques that are shown in the financial outputs below, are used to measure the relationships among the financial statement items of a single set of statements and the changes that have taken place in these items as reflected by successive financial statements. The objective of any analytical method or model is to simplify the data under review to more understandable terms.

The personal aspects of the case are not discussed, the intent of the case is to present the financial data so that a sample execution of the model can be shown. In the next few pages the actual print-out that the banker would receive on Limber Lumber Company is presented.

After performing "login" to CPS and after loading the program, execution can begin. By typing in "xeg" program, execution can be initiated.

The first instruction given by the model tells the user to enter the number of years of historical financial statements that he wishes to analyze. After entering the number of years, the program requests balance sheet data and income statement data for each year. Sample execution of the first part of the program is shown below.

```
?login(bus1,12243)
PASSWORD:
********
GOOD EVENING; USER 08; TIME  18:44:44 12/04/75;
?load(Dave)
?xeq


This is a program to be used in a Credit Analysis of any business loan request.

The user will be asked to enter information into the computer.

This information will be requested in the form of summary accounts.  Listed below is a list
of the accounts used by the program.  Please arrange your financial data to fit into these acc


ACCOUNT GROUPINGS  ABBREVIATIONS USED

CASH -------------------- CASH
ACCOUNT RECEIVABLES --- A/R
INVENTORIES ------------ INV
OTHER CURRENT ASSETS -- OCASET
NET FIXED ASSETS ------ FASSET
ACCOUNT PAYABLE ------- A/P
OTHER CURRENT LIAB ---- OCLIAB
LONG-TERM LIAB -------- LTLIAB
COMMON STOCK ---------- CSTK
PERFERRED STOCK ------- PFD
RETAIN EARNINGS ------- RETAIN

SALES ------------------ SALES
COST OF GOODS SOLD ---- CGS
TOTAL OPERATING EXP --- TOX
INTEREST EXPENSE ------ INTEXP
TAXES ------------------ TAXES

Please enter this information as it is asked for by the computer.

The user will be asked yes and no questions, please answer in capital letters.


Enter the number of past years of financial data to be analyzed.(max 5 yrs)
yrs
?5


Enter the Balance Sheet Data for years 1

cash(1)
?31000
inv(1)
?542000
ar(1)
?136000
ocaset(1)
?23000
fasset(1)
?337000
ap(1)
?107000
ocliab(1)
?
ltlib(1)
?0
cstk(1)
?216000
pfd(1)
?0
retain(1)
?691000
```

```
Now enter the Income Statement Data for year 1

sales(1)
?1542000
cgs(1)
?1103000
tox(1)
?391000
intexp(1)
?16000
taxes(1)
?10000


Enter the Balance Sheet Data for years 2

cash(2)
?24000
inv(2)
?641000
ar(2)
?114000
ocaset(2)
?20000
fasset(2)
?316000
ap(2)
?71000
ocliab(2)
?20000
ltlib(2)
?106000
cstk(2)
?216000
pfd(2)
?0
retain(2)
?702000

Now enter the Income Statement Data for year 2

sales(2)
?1600000
cgs(2)
?1193000
tox(2)
?418000
intexp(2)
?9000
taxes(2)
?0


Enter the Balance Sheet Data for years 3

cash(3)
?19000
inv(3)
?646000
ar(3)
?170000
ocaset(3)
?10000
fasset(3)
?292000
ap(3)
?71000
ocliab(3)
?21000
ltlib(3)
?146000
cstk(3)
?216000
pfd(3)
?0
retain(3)
?683000
```

```
Now enter the Income Statement Data for year 3

sales(3)
?1400000
cgs(3)
?998000
tax(3)
?375000
intexp(3)
?12000
taxes(3)
?0


Enter the Balance Sheet Data for years 4

cash(4)
?106000
inv(4)
?556000
ar(4)
?152000
ocaset(4)
?10000
fasset(4)
?329000
ap(4)
?82000
ocliab(4)
?24000
ltlib(4)
?132000
cstk(4)
?216000
pfd(4)
?0
retain(4)
?674000


Now enter the Income Statement Data for year 4

sales(4)
?1800000
cgs(4)
?1250000
tax(4)
?400000
intexp(4)
?13000
taxes(4)
?15000


Enter the Balance Sheet Data for years 5

cash(5)
?40000
inv(5)
?557000
ar(5)
?250000
ocaset(5)
?15000
fasset(5)
?350000
ap(5)
?172000
ocliab(5)
?35000
ltlib(5)
?146000
cstk(5)
?216000
pfd(5)
?0
retain(5)
?637000
```

```
Now enter the Income Statement Data for year 5

sales(5)
?2000000
cgs(5)
?1400000
tox(5)
?430000
intexp(5)
?14000
taxes(5)
?41000
```

The first output that the loan officer receives is the historical balance sheet, income and cash flow statements. The statements provide the loan officer with the means of checking his inputs of historical data. The compact arrangement of these statements provide the loan officer with five years of historical data on less than one page. The loan officer can quickly evaluate these statements to determine changes in individual accounts from one year to the next.

```
To determine the cash flow you must enter the amount of depriciation taken in the preceding y

depr(1)
?19000
depr(2)
?26000
depr(3)
?28000
depr(4)
?26000
depr(5)
?29000
```

he computer will now print-out the information just read into the computer showing Balance Shee
ncome Statements and Cash Flow Statements.

Period=	1	2	3	4	5
CASH	31000.00	24000.00	19000.00	106000.00	34000.00
INVENT	542000.00	641000.00	646000.00	531000.00	557000.00
A/R	136000.00	114000.00	170000.00	152000.00	250000.00
OCASSET	23000.00	20000.00	10000.00	10000.00	15000.00
TOTCA	732000.00	799000.00	845000.00	799000.00	856000.00
FASSET	337000.00	316000.00	292000.00	329000.00	350000.00
TOTASS	1069000.00	1115000.00	1137000.00	1128000.00	1206000.00
A/P	107000.00	71000.00	71000.00	82000.00	172000.00
OCLIAB	55000.00	20000.00	21000.00	24000.00	35000.00
TOTCT	162000.00	91000.00	92000.00	106000.00	207000.00
LTLIAB	0.	106000.00	146000.00	132000.00	146000.00
CSTK	216000.00	216000.00	216000.00	216000.00	216000.00
PFD	0.	0.	0.	0.	0.
RETAIN	691000.00	702000.00	683000.00	674000.00	637000.00
T NW/L	1069000.00	1115000.00	1137000.00	1128000.00	1206000.00
SALES	1542000.00	1600000.00	1400000.00	1800000.00	2000000.00
CGS	1103000.00	1193000.00	998000.00	1250000.00	1400000.00
MARGIN	439000.00	407000.00	402000.00	550000.00	600000.00
OP EXP	391000.00	418000.00	375000.00	400000.00	430000.00
EBIT	48000.00	-11000.00	27000.00	150000.00	170000.00
INT EXP	16000.00	9000.00	12000.00	13000.00	14000.00
EBT	32000.00	-20000.00	15000.00	137000.00	156000.00
TAXES	10000.00	0.	0.	15000.00	41000.00
NI	22000.00	-20000.00	15000.00	122000.00	115000.00
NI	22000.00	-20000.00	15000.00	122000.00	115000.00
DEPR	19000.00	26000.00	28000.00	26000.00	29000.00
CHANGE TOTCA		74000.00	51000.00	-133000.00	129000.00
CHANGE TOTCT		-71000.00	1000.00	14000.00	101000.00
CASH FLOW		-139000.00	-7000.00	295000.00	116000.00

The next three statements, consisting of comparative balance sheets and income statements using percentage relationships, provide the user with an indepth analysis of historical financial trends. The first statement printed by the computer is a historical balance sheet showing the relationship between individual accounts and sales in that year. The last statement in this section consists of a percentage balance sheet which shows the relationship between total assets and other asset accounts and total net worth plus liabilities in relation to liabilities, common stock and retain earnings.

Do you wish for the computer to print-out the percentage Balance Sheet and Income Statemen
?YES

as a percentage of Sale

Period=	1	2	3	4	5	AVE
CASH	.020	.015	.014	.059	.017	.025
INVENT	.351	.401	.461	.295	.278	.357
A/R	.088	.071	.121	.084	.125	.098
OCASSET	.015	.012	.007	.006	.007	.010
TOTCA	.475	.499	.604	.444	.428	.490
FASSET	.219	.197	.209	.183	.175	.196
TOTASS	.693	.697	.812	.627	.603	.686
A/P	.069	.044	.051	.046	.086	.059
OCLIAB	.036	.012	.015	.013	.017	.019
TOTCT	.105	.057	.066	.059	.103	.078
LTLIAB	0.	.066	.104	.073	.073	.063
CSTK	.140	.135	.154	.120	.108	.155
PFD	0.	0.	0.	0.	0.	0.
RETAIN	.448	.439	.488	.374	.318	.414
T NW/L	.693	.697	.812	.627	.603	.812

SALES	1.000	1.000	1.000	1.000	1.000	1.000
CGS	.715	.746	.713	.694	.700	.714
MARGIN	.285	.254	.287	.306	.300	.286
OP EXP	.254	.261	.268	.222	.215	.244
EBIT	.031	-.007	.019	.083	.085	.042
INT EXP	.010	.006	.009	.007	.007	.008
EBT	.021	-.012	.011	.076	.078	.035
TAXES	.006	0.	0.	.008	.020	.007
NI	.014	-.012	.011	.068	.057	.028

Do you wish to see a print-out of the comparative Balance Sheet comparing the accounts to Total Assets and Total Net Worth+Liabilities?
?YES

Periods=	1	2	3	4	5	AVE
CASH	.029	.022	.017	.094	.028	.038
INVENT	.507	.575	.568	.471	.462	.517
A/R	.127	.102	.150	.135	.207	.144
OCASSET	.022	.018	.009	.009	.012	.014
TOTCA	.685	.717	.743	.708	.710	.713
FASSET	.315	.283	.257	.292	.290	.287
TOTASS	1.000	1.000	1.000	1.000	1.000	1.000
A/P	.100	.064	.062	.073	.143	.088
OCLIAB	.051	.018	.018	.021	.029	.028
TOTCT	.152	.082	.081	.094	.172	.116
LTLIAB	0.	.095	.128	.117	.121	.092
CSTK	.202	.194	.190	.191	.179	.191
PFD	0.	0.	0.	0.	0.	0.
RETAIN	.646	.630	.601	.598	.528	.600
T NW/L	1.000	1.000	1.000	1.000	1.000	1.000

A ratio analysis comes next. The computer prints sixteen ratios for each year of historical data. These ratios give the user the means for a quick analysis of the profitability, activity, debt leverage and liquidity of the firm. By comparing these ratios to industry standards, the loan officer is able to determine the relative financial and operating strengths of the firm being analyzed.

ou wish to see a printout of the ratio analysis.

Periods=	1	2	3	4	5
IQUIDITY RATIO					
nt-Ratio	4.519	8.780	9.185	7.538	4.135
-Ratio	1.173	1.736	2.163	2.528	1.444
o W.C.	.951	.905	.858	.766	.858
EVERAGE RATIO					
to Assets	.152	.177	.209	.211	.293
P to EBIT	3.000	-1.222	2.250	11.538	12.143
CTIVITY RATIO					
ily Sales	32.192	26.006	44.321	30.822	45.625
ab to Net Wort	.179	.099	.102	.119	.243
to Net Worth	.372	.344	.325	.370	.410
Turnover	4.576	5.063	4.795	5.471	5.714
Turnover	1.442	1.435	1.231	1.596	1.658
Velocity	49.742	66.667	73.684	16.981	58.824
urnover	2.845	2.496	2.167	3.390	3.591
ROFITABILITY RATIOS					
t Margin	.014	-.012	.011	.068	.057
ting Exp/Sales	.254	.261	.268	.222	.215
n on Assets	.021	-.018	.013	.108	.095

The computer next provides the loan officer with a report of balance sheet changes between each period of historical data. From this the loan officer is able to determine if funds have been appropriately applied in the past.

YEAR 1 to 2	SOURCES	USES
CASH	7000.00	
INVENT		99000.00
A/R	22000.00	
OCASSET	3000.00	
FASSET	21000.00	
A/P		36000.00
OCLIAB		35000.00
LTLIAB	106000.00	
CSTK		
PFD		
RETAIN	11000.00	
TOTAL	170000.00	170000.00

YEAR 2 to 3	SOURCES	USES
CASH	5000.00	
INVENT		5000.00
A/R		56000.00
OCASSET	10000.00	
FASSET	24000.00	
A/P		
OCLIAB	1000.00	
LTLIAB	40000.00	
CSTK		
PFD		
RETAIN		19000.00
TOTAL	80000.00	80000.00

YEAR 3 to 4	SOURCES	USES
CASH		87000.00
INVENT	115000.00	
A/R	18000.00	
OCASSET		
FASSET		37000.00
A/P	11000.00	
OCLIAB	3000.00	
LTLIAB		14000.00
CSTK		
PFD		
RETAIN		9000.00
TOTAL	147000.00	147000.00

YEAR 4 to 5	SOURCES	USES
CASH	72000.00	
INVENT		26000.00
A/R		98000.00
FASSET		21000.00
A/P	90000.00	
OCLIAB	11000.00	
LTLIAB	14000.00	
CSTK		
PFD		
RETAIN		37000.00
TOTAL	187000.00	187000.00

The initial outputs are historical in nature. The next three procedures use this historical data to produce two different types of projected balance sheets and a pro-forma income statement. Pro-forma statements are a means to potentially judge future credit-worthiness of the borrower. Financial data predicting future results and trends should not be considered as true predictor of the future, but should serve only as a guide. These pro-forma statements may assume relationships which are not true, (e.g., continued growth of common stock at the same rate as in the past). The loan officer should be able to recognize rough areas and make the proper adjustments. The main purpose of pro-forma statements is to provide a starting point for predicting the borrower's ability to repay loan commitments.

The first pro-forma statement presented in this model is a projected balance sheet for the next five years. The means used to calculate this statement is a projection of future sales either by using the borrower's estimate of future sales, or the computer projections using regression analysis of past sales trends. Once the future sales are determined, the individual accounts are calculated by assuming a constant relationship between the average historical sales and balance sheet totals.

The programs listed below are part of the analysis but because of techincal constraints limi
on the OSU computer system you must load and excute these programs yourself. Follow the ins
have no problems.

Types of Additional Pragrams	Keys needed in Loading
Proforma Balance Sheets(Sales Relationship)	Dave85
Proforma Balance Sheets(Asset and N Worth Relationship)	Dave87
Proforma Income Statements	Dave9
Monthly Cash Budget	Dave93

The programs that you desire to to excute load in the folowing manner.

Type in this statement ----- load(Dave93)
Wait for the computer to answer then type in xeq

For each additional program you wish to use Just type in the key for loading and xeq to star
?

```
 load(dave85,1b)
** KEY INCORRECT.
?load(dave85)
?xeq
```

Are the Sales projections printed below from a regression of your past sales a accurate
forecast of future sales. If they are not the computer will ask for your own projections.

```
Salpro        2003200.00     2114800.00     2226400.00     2338000.00     2449600.00
?YES
```

Balance Sheet Projections based on Sales relationship for Years

	1	2	3	4	5
CASH	49905.35	52685.62	55465.89	58246.16	61026.43
INVENY	715961.77	755848.61	795735.46	835622.31	875509.16
A/R	196441.87	207385.82	218329.76	229273.71	240217.65
OCASSET	19076.12	20138.86	21201.61	22264.36	23327.10
TOTCA	981385.10	1036058.91	1090732.72	1145406.54	1200080.35
FASSET	393587.35	415514.44	437441.53	459368.63	481295.72
TOTASS	1374972.46	1451573.36	1528174.26	1604775.16	1681376.06
A/P	118603.56	125211.07	131818.57	138426.08	145033.59
OCLIAB	37660.68	39758.79	41856.90	43955.01	46053.12
TOTCT	156264.24	164969.86	173675.47	182381.09	191086.70
LTLIAB	126950.42	134022.93	141095.45	148167.97	155240.48
CSTK	263366.13	278038.48	292710.83	307383.19	322055.54
PFD	0.	0.	0.	0.	0.
RETAIN	828391.67	874542.09	920692.50	966842.92	1012993.33
T NW/L	1374972.46	1451573.36	1528174.26	1604775.16	1681376.06

?

Next, the computer provides the user with another pro-forma balance sheet. This statement assumes a constant relationship between individual assets and total assets. A similiar relationship is assumed between individual liabilities and total net worth plus liabilities. The computer projects a growth pattern of total asset and net worth plus liabilities, by using regressional analysis. If these projections are not considered appropriate, the user may enter other projections.

These two different types of projected balance sheets should help the loan officer determine the amount of permanent working capital that is needed as sales increase.

```
load(dave87,1b)

** KEY INCORRECT.
?load(Dave87,1b)
?xeq

Presented below are the growth patterns of your Total Assets and Net Worth.  If they are not c
the computer will ask for your own projection.

TOTAL ASSET Projection      1217100.00      1245800.00      1274500.00      1303200.00      1331900.(
?YES

Balance Sheet Projections based on a Total Asset and Total NW+liability relationship for the Y
                    1               2               3               4               5

CASH              46103.35        47190.50        48277.65        49364.79        50451.94
INV              628672.97       643497.48       658321.99       673146.51       687971.02
A/R              175512.82       179651.52       183790.23       187928.93       192067.64
OCASET            16930.05        17329.27        17728.50        18127.72        18526.94
TOTCA            867219.19       887668.78       908118.36       928567.95       949017.53
FASSET           349880.81       358131.22       366381.64       374632.05       382882.47
TOTASS          1217100.00      1245800.00      1274500.00      1303200.00      1331900.00

A/P              107477.47       110011.86       112546.25       115080.63       117615.02
OCLIAB            33629.69        34422.70        35215.71        36008.72        36801.72
TOTCT            141107.16       144434.55       147761.95       151089.35       154416.75
LTLIAB           112352.45       115001.79       117651.13       120300.48       122949.82
CSTK             232794.09       238283.52       243772.96       249262.39       254751.83
PFD                     0.              0.              0.              0.              0.
RETAIN           730846.31       748080.13       765313.96       782547.78       799781.61
T NW/L          1217100.00      1245800.00      1274500.00      1303200.00      1331900.00

?
```

At this point of execution, the model presents a pro-forma income statement and cash flow statements. The projected net income is calculated by assuming a constant relationship between expenses and sales. The projected sales in this procedure are determined by regressional analysis of past sales, or if these projections are not accurate, the user may enter his own projections. The procedure next projects the cash flow for five future years. The computer then derives the amount of cash available after current debt payments and proposed debt payments are paid. This statement helps the loan officer analyze projected cash flows to determine if the company can generate enough cash to repay the new loan along with current loan obligations.

Before the program can execute the user must enter certain data. The user must enter the amount of the loan requested, the interest rate and the length of time that the funds are to be borrowed. Procedure also requires a knowledge of current debt payments and the amount of projected depreciation to be taken for the next five years.

```
load(Dave@1,lb)
* PROGRAM NOT FOUND.

@ed&dave91,lb)
xeq

ere the Sales Projections from the regression accurate?  If not the computer will ask for the c
YES

                                                                                    Sales Forcas

ter the amount of annual debt payments due of the current long term loans for the next 5 years
lt(1)
18000
:t(2)
!0000
 t(3)
!0000
 t(4)
!0000
 t(5)
!1000
```

```
Enter the amount of new debt requested.
db
?100000

Enter the interest rate and the term of the new loan.
R
?.13
Pb
350

Please enter the approx. amount of deprication to be taken in the next five years.
depr(1)
?30000
depr(2)
?30000
depr(3)
?30000
depr(4)
?29000
depr(5)
?29000
```

Projected Income for Years

	1	2	3	4	5
SALPRO		2003200	2114800	2226400	2338000	2449600
CGS		1429580	1509220	1588860	1668500	1748150
MARGIN		573624	605581	637538	669495	701452
OP EXP		488739	515967	543195	570423	597651
EBIT		84885	89614	94343	99072	103801
INTEXP		14000	14000	14000	14000	14000
EBT		70885	75614	80343	85072	89801
TAXES		14150	14938	15727	16515	17303
NI		56735	60675	64616	68557	72497

NI	56735	60675	64616	68557	72497
DEPRICATION	30000	30000	30000	29000	29000
CHANGE TOTCA	109480	51894	51894	51893	51893
CHANGE TOTCT	-50736	8706	8706	8706	8706
CASH FLOW	-73481	47487	51428	54369	58310
CASH AVA FOR DEBT PAYMENTS AND DIVIDENDS	-73481	47487	51428	54369	58310
CURRENT DEBT PAYMENT	18000	20000	20000	20000	21000
NEW DEBT PAYMENT	26500	26500	26500	26500	26500
INTEXP ON NEW DEBT	13000	11245	9262	7021	4489
PRINCIPLE ON NEW DEBT	13500	15255	17238	19479	22011
CASH AVAIABLE AFTER DEBT PAYMENTS	-111481	6610	9559	11379	13054

```
?
```

The last output of the model is a projected cash budget for the next year. After the computer has compared monthly revenue and monthly operating expenses, the model prints an orderly statement showing the monthly cash needs of the business. This statement reflects the seasonal nature of the business and whether a line of credit is needed to meet monthly operating payments. These monthly deficits, if any, should be financed with short-term funds, and should always be repaid from conversion of working capital.

```
load(Dave93)
xeq

e will now perform a monthly Cash Budget.  You must enter certain new data, consisting of Montl
onths plus the monthly sales for the last 2 months just completed, Total Operating Expenses anc
utlays.  These accounts will be defined below.

onthly Sales(monsal)consists of your projected monthly sales of the next 12 month period.
otal Operating Expenses(tox) consists of purchases of material, salaries, labor expenses and ot
ther Monthly Expenses(otpay) consists of taxes, dividends and repayment of current debt.


hat is your interest rate on the line of credit.

.13


lease enter the last 2 months sales first and then enter the projected sales for the next 12 mc
onsal(1)
60000
onsal(2)
60000
onsal(3)
60000
onsal(4)
70000
onsal(5)
105000
onsal(6)
150000
onsal(7)
250000
onsal(8)
351000
onsal(9)
400000
onsal(10)
200000
onsal(11)
185000
onsal(12)
?9000
onsal(13)
?5000
onsal(14)
?5000

lease enter the monthly Cash Operating outlays projected for the next 12 months as defined abov
ntox(1)
60000
ntox(2)
80000
ntox(3)
.20000
ntox(4)
.30000
ntox(5)
?55000
ntox(6)
?75000
ntox(7)
ntox(8)
.90000
ntox(9)
50000
ntox(10)
5000
ntox(11)
5000
ntox(12)
5000
```

,er Other monthly Cash Payments in the month to be paid.
'ay(1)

'ay(2)

ay(3)
00
ay(4)

ay(5)

ay(6)
000
ay(7)

ay(8)

ay(9)
)00
ay(10)

ay(11)

ay(12)
)00

ed to know the minimun cash balance desired.
ah
)00

, percentage of Total Operating Expenses are allocated to material purchases.
h

percentage of sales are collected in first month of sales, second month and third month.
ct(1)

ct(2)

ct(3)

r your beginning cash balance.
al
)0

	1	2	3	4	5	6	7	8	9	10	11	12	TOTAL
MONTHLY SALES	60000	70000	105000	150000	250000	351000	400000	200000	185000	99000	75000	75000	2020000
COLLECTIONS													
FIRST MONTH .700	42000	49000	73500	105000	175000	245700	280000	140000	129500	69300	52500	52500	
SECOND MONTH .200	12000	12000	14000	21000	30000	50000	70200	80000	40000	37000	19800	15000	
THIRD MONTH .100	6000	6000	6000	7000	10500	15000	25000	35100	40000	20000	18500	9900	
TOTAL	60000	67000	93500	133000	215500	310700	375200	255100	209500	126300	90800	77400	2013999
PURCHASES	48000	64000	96000	104000	204000	300000	260000	152000	120000	68000	52000	52000	1519999
OF EXPENSES	12000	16000	24000	26000	51000	75000	65000	38000	30000	17000	13000	13000	380000
OTHER PAYMENTS	0	0	5000	0	0	50000	0	0	20000	0	0	25000	100000
NET CASH GAIN(LOSS)	0	-13000	-31500	3000	-39500	-114300	50200	65100	39500	41300	25800	-12600	
BEGINNING BALANCE	14000	14000	1000	-30500	-27500	-67000	-181300	-131100	-66001	-26501	14799	40599	
CUMMULATIVE CASH	14000	1000	-30500	-27500	-67000	-181300	-131100	-66001	-26501	14799	40599	27999	
DESIRED LEVEL OF CASH	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	
INTEXP ON LINE OF CREDIT	-11	-152	-493	-460	-888	-2127	-1583	-878	9	-2	0	0	
TOTAL LOANS NEEDED OR SURPLUS	-1011	-14152	-45993	-42961	-82889	-198427	-147683	-81878	-41950	-203	25599	12999	
CASH FLOW FROM OPERATIONS													6957

?

logout(acct)

When the program has completed execution, the loan officer has an analysis of historical financial statements to examine for trends and proper financing patterns. Pro-forma financial information is also provided to help predict the future. The monthly cash budget information can give the loan officer an idea of the type and amount of financing needed for current assets. With this system of inputs and outputs, the loan officer has an organized set of aids to assist in his credit decision process.

The loan officer can infer from the sample case (Limber Lumber) that it is questionable, whether the $100,000 loan should be made. Limber Lumber will just barely generate enough cash to meet the future debt payments, if the loan is made. Closer analysis must be made in this situation, by the loan officer. The monthly cash budget sheet reflects that a line of credit of $151,042 is needed to meet seasonal needs. It appears the line of credit should possible be extended to $175,000 to give Limber Lumber some flexibility in their cash management.

The overall evaluation of the firm reveals that it is a fairly sound business, which has some problems in cash management policy and there appears to be a need for increased permanent working capital. A term loan should be made to Limber Lumber, but the amount should be reduced from the $100,000 total so that the company will be able to meet debt payment.

While there are many qualitative and behaviorial factors involved in any credit decision, the model presented in the above discussion can help with many of the analytical aspects in judging credit-worthiness. Such models would seem to have considerable potential in the world of

commercial bank lending to facilitate lending decisions.

Summary

The model is designed to function as an aid to the loan officers in their credit analysis of customers. The programs that made up the model are written in CPS/PL1 for use on an IBM 360 computer. The time sharing aspect of the model makes it economically feasible for banks to use. The model enables the loan officer to utilize the power and speed of the computer in a conversational manner. Sufficient instructions are provided, so that once loaded and executed, the model "leads" the user through completion. Only a minimal amount of computer knowledge is required to operate this model.

The fast and complete analysis of loan customers at a reasonable cost will enhance the profit objectives of banks. This model should provide one of the tools that are needed by bankers to improve the quality of their loan portfolios.

CHAPTER V

CONSLUSION AND SUMMARY

Banks must develop greater sophistication in their credit analysis and decision procedures, if they desire to maintain the same levels of profits as in past years. The risk associated with commercial lending has increased in recent years, mainly because of the instability and complexity of the environment, the longer term of loans and the higher cost of funds.

The increased risk requires indepth analysis of loan customers, but because of the rapidly changing environment credit decisions must be made rapidly. The means for banks to solve this dilema is to develop computer models.

This project was undertaken to create a model to assist banks in their commercial loan decisions. The model provides the loan officer with an orderly arrangement of the financial data needed to perform a credit analysis on a business. Final evaluation of the model developed in this paper will depend on actual usage in a number of trial credit analysis by loan officers. The effectiveness of the model can only be determined by testing the model in actual credit decisions.

BIBLIOGRAPHY

1. Ostlund, Dow. "Commerical Lending Today and Tomorrow". Banker's Monthly. August 15, 1972, p. 9.

2. Ibid. pp. 9-10.

3. Vandell, Robert F., and Robert M. Pennell. "Tight Money Financing". Harvard Business Review. September-October 1971, p. 97.

4. Houghet, George. "Techniques of Term Loan Analysis". The Journal of Commerical Bank Lending. March 1975, pp. 30-36.

5. Anderlik, John. "How Bankers Make Their Credit Decisions". Credit and Financial Management. August 1974, pp. 14-37.

6. Hunn, Paul. "Bank Credit in the 1970's - New Realities and Old Verities". The Journal of Commerical Bank Lending. June 1971, pp. 29-35.

7. Cooper, P.S. "Four P's for Lending". The Journal of Commerical Bank Lending. July 1975, pp. 46-51.

8. Kujawa, Daniel. "A Banker's Approach to Business Loans". The Journal of Commerical Bank Lending. December 1971, p. 37.

9. Ensign, James. "The Elements of Unsecured Business Loans". The Journal of Commerical Bank Lending. January 1973, pp. 18-33.

10. Ibid. p. 20

11. Hunn. pp. 29-35.

12. Kajawa. pp. 33-47.

13. Ensign. pp. 18-33

14. Hunn. pp. 29-35

15. Kujawa. pp. 33-47.

16. Kujawa. p. 40.

17. Ibid. p. 41.

18. Hunn. p. 31.

19. Johnson, Robert W., Financial Management. Boston Allyn and Bam, Inc., 1971, pp. 214-216.

20. Ibid. p. 221

21. Weston, J., and Eugene Brigham, Managerial Finance. 4th Edition, New York, Holt Rinehart and Winston, pp. 88-106.

22. Kujawa. p. 39.

23. Ibid. p. 41

24. Waterman, Robert and Robert Gee. "A New Tool for Bank Management". The Journal of Commerical Bank Lending. September 1967, pp. 80-98.

25. Orgler, Y.E. "Selection of Bank Loans for Evaluation: An Analytical Approach". Journal of Finance. March 1969, pp. 75-83.

26. Eilon, D., and T.R. Fowles. Applications of Management Science in Banking and Finance. London's Grower Press, 1972.

27. Morris, R.A. "Credit Analysis: An O.R. Approach". The Journal of Commerical Bank Lending. July 1966, pp. 626-638.

28. Williams, S. "Survey Shows Increasing Importance of Mathematical Aids to Management". Journal of Bank Research. Summer 1970, pp. 6-18.

29. Waterman and Gee. pp. 80-99.

30. Myers, James and Edward Forgy. "The Development of Number Credit Evaluation Systems". The Journal of American Statistics. September 1963, pp. 789-826.

31. Hammer, F.S., and Y.E. Orgler. "Developments in Credit Scoring for Commerical Loans". Journal of Commerical Bank Lending. July 1969, pp. 25-31.

32. Heffenhouse, George and Jack Wentworth. "A New Look for Credit Scoring". The Journal of Commerical Bank Lending. December 1971, pp. 26-34.

33. Ewert, D.C. "Trade Credit Management: Selection of Accounts Receivable Using a Statistical Model". The Journal of Commerical Bank Lending. September 1969, pp. 71-78.

34. Edmister, R.O. "Financial Ratios and Credit Scoring for Small Business Loans". The Journal of Commerical Bank Lending. September 1971, pp. 10-23.

35. Edmister, R.O. "An Empircal Test of Financial Ratio Analysis for Small Business Failure Prediction". Journal of Financial and Quantitative Analysis. March 1972, pp. 2477-2493.

36. Abate, R.P. "Numerical Scoring Systems for Commerical Loans". Bankers Monthly. January 1969, pp. 38-44.

37. Cohen, K.J., T.C. Gilmore and F.A. Singer. "Bank Procedures for Analyzing Business Loan Applications". Analytical Methods in Banking. K.J. Cohen and F.S. Hamner (eds.) Homewood, Ill.: Richard D. Irwin, 1966, pp. 218-251.

38. Heffenhouse and Wentworth. p. 26.

39. Sangster, Bruce F. and Sam Raguso, Jr. "An Implementation of Computer Credit Analysis". The Journal of Commerical Bank Lending. March 1971, pp. 55-63.

40. Burd, Richard and Charles Blades. "Automating Commerical Loan Analysis". Burroughs Clearing House. January 1974, pp. 26-33.

APPENDIX

LISTING OF COMPUTER PROGRAMS

```
 5.            DECLARE  Dave1 ENTRY  EXT;
10.            DECLARE  Dave2a ENTRY  EXT;
11.            DECLARE  Dave2b ENTRY  EXT;
12.            DECLARE  Dave2c ENTRY  EXT;
12.5           DECLARE  Dave2d ENTRY  EXT;
13.            DECLARE  Dave25 ENTRY  EXT;
15.            DECLARE  Dave3 ENTRY  EXT;
20.            DECLARE  Dave4 ENTRY  EXT;
22.            DECLARE  Dave85 ENTRY  EXT;
22.5           DECLARE  Dave87 ENTRY  EXT;
24.            DECLARE  Dave93 ENTRY  EXT;
24.1           DECLARE  Dave9 ENTRY  EXT;
25.            CALL Dave1;
30.            RELEASE ;
30.1           DECLARE depr(5);
31.            PUT LIST('The computer will now print-out the information just read into the computer showing Balance Sheets, N ');
31.1           PUT LIST('Income Statements and Cash Flow Statements.');
35.            CALL Dave2a(depr);
37.            RELEASE ;
38.            CALL Dave2b(depr);
39.            RELEASE ;
39.05  hot:    ;
39.1           PUT LIST('Do you wish for the computer to print-out the percentage Balance Sheet and Income Statement as a percenta
39.2           READ INTO(ans) ;
39.3           IF upcase(ans)='YES' THEN GO TO heat;
39.4           IF upcase(ans)='NO' THEN GO TO there; ELSE GO TO hot;
40.    heat:   CALL Dave2c;
41.            RELEASE ;
42.            CALL Dave2d;
43.    there:  RELEASE ;
44.            CALL Dave25;
44.5           RELEASE ;
45.            CALL Dave3;
50.            RELEASE ;
55.            CALL Dave4;
60.            RELEASE ;
60.05          PUT LIST('');
65.            PUT LIST('The programs listed below are part of the analysis but because of techincal constraints limiting to size
66.            PUT LIST('on the OSU computer system you must load and excute these programs yourself.  Follow the instruction belo
67.            PUT LIST('have no problems.');
68.            PUT LIST('');
69.            PUT LIST('');
70.            PUT LIST('          Types of Additional Programs                    Keys needed in Loading');
71.            PUT LIST('');
72.            PUT LIST('Proforma Balance Sheets(Sales Relationship)                    Dave85');
73.            PUT LIST('Proforma Balance Sheets(Asset and N Worth Relationship)        Dave87');
74.            PUT LIST('Proforma Income Statements                                     Dave9');
75.            PUT LIST('Monthly Cash Budget                                            Dave93');
76.            PUT LIST('');
77.            PUT LIST('The programs that you desire to to excute load in the folowing manner.');
78.            PUT LIST('');
79.            PUT LIST('Type in this statement ----- load(Dave93)');
80.            PUT LIST('Wait for the computer to answer then type in xeq');
81.            PUT LIST('');
82.            PUT LIST('For each additional program you wish to use just type in the key for loading and xeq to start the program
?
```

```
 15.          ;
 20.          DECLARE  @ball FILE OUTPUT ENV( V(120)  ) ;
 25.          DECLARE cash(5) DEC(6);
 30.          DECLARE inv(5) DEC(6);
 35.          DECLARE ar(5) DEC(6);
 40.          DECLARE ocaset(5) DEC(6);
 45.          DECLARE fasset(5) DEC(6);
 55.          DECLARE ap(5) DEC(6);
 60.          DECLARE ocliab(5) DEC(6);
 70.          DECLARE ltlib(5) DEC(6);
 80.          DECLARE cstk(5) DEC(6);
 85.          DECLARE pfd(5) DEC(6);
 90.          DECLARE retain(5) DEC(6);
100.          DECLARE sales(5) DEC(6);
105.          DECLARE cgs(5) DEC(6);
115.          DECLARE tox(5) DEC(6);
125.          DECLARE intexp(5) DEC(6);
135.          DECLARE taxes(5) DEC(6);
160.          PUT LIST('');
165.          PUT LIST('');
170.          PUT LIST('');
175.          PUT LIST('This is a program to be used in a Credit Analysis of any business loan request.');
175.1         PUT LIST('');
176.          PUT LIST('The user will be asked to enter information into the computer.');
176.1         PUT LIST('');
177.          PUT LIST('This information will be requested in the form of summary accounts.  Listed below is a list');
178.          PUT LIST('of the accounts used by the program.  Please arrange your financial data to fit into these accounts.');
178.4         PUT LIST('');
178.45        PUT LIST('');
178.5         PUT LIST('');
178.51        PUT LIST('ACCOUNT GROUPINGS  ABBREVIATIONS USED');
178.52        PUT LIST('');
178.53        ;
178.56        PUT LIST('CASH ------------------- CASH');
178.59        PUT LIST('ACCOUNT RECEIVABLES --- A/R');
178.62        PUT LIST('INVENTORIES ------------ INV');
178.65        PUT LIST('OTHER CURRENT ASSETS -- OCASET');
178.68        PUT LIST('NET FIXED ASSETS ------- FASSET');
178.71        PUT LIST('ACCOUNT PAYABLE -------- A/P');
178.74        PUT LIST('OTHER CURRENT LIAB ----- OCLIAB');
178.77        PUT LIST('LONG-TERM LIAB --------- LTLIAB');
178.8         PUT LIST('COMMON STOCK ----------- CSTK');
178.83        PUT LIST('PERFERRED STOCK -------- PFD');
178.86        PUT LIST('RETAIN EARNINGS -------- RETAIN');
178.89        PUT LIST('');
178.92        PUT LIST('SALES ------------------ SALES');
178.95        PUT LIST('COST OF GOODS SOLD ----- CGS');
178.98        PUT LIST('TOTAL OPERATING EXP ---- TOX');
179.01        PUT LIST('INTEREST EXPENSE ------- INTEXP');
179.04        PUT LIST('TAXES ------------------ TAXES');
180.          PUT LIST('');
185.          PUT LIST('Please enter this information as it is asked for by the computer.');
190.          PUT LIST('');
195.          PUT LIST('The user will be asked yes and no questions, please answer in capital letters.');
200.          PUT LIST('');
205.          PUT LIST('');
210.          PUT LIST('Enter the number of past years of financial data to be analyzed.(max 5 yrs)');
215.          GET LIST(yrs);
```

```
265.            PUT LIST('Now enter the Income Statement Data for year',i);
270.            PUT LIST('');
275.            GET LIST(sales(i),cgs(i),tax(i),intexp(i),taxes(i));
280.            PUT LIST('');
285.            PUT LIST('');
290.            PUT LIST('');
295.            PUT LIST('');
300.            END ;
355.            OPEN FILE(@ball) OUTPUT ;
360.            WRITE FILE(@ball) FROM(yrs) ;
365.            WRITE FILE(@ball) FROM(cash) ;
370.            WRITE FILE(@ball) FROM(inv) ;
375.            WRITE FILE(@ball) FROM(ar) ;
380.            WRITE FILE(@ball) FROM(ocaset) ;
390.            WRITE FILE(@ball) FROM(fasset) ;
400.            WRITE FILE(@ball) FROM(ap) ;
405.            WRITE FILE(@ball) FROM(ocliab) ;
415.            WRITE FILE(@ball) FROM(ltlib) ;
425.            WRITE FILE(@ball) FROM(cstk) ;
430.            WRITE FILE(@ball) FROM(pfd) ;
435.            WRITE FILE(@ball) FROM(retain) ;
445.            WRITE FILE(@ball) FROM(sales) ;
450.            WRITE FILE(@ball) FROM(cgs) ;
460.            WRITE FILE(@ball) FROM(tax) ;
470.            WRITE FILE(@ball) FROM(intexp) ;
480.            WRITE FILE(@ball) FROM(taxes) ;
490.            END Dave1;
?
```

```
  5.02          PUT LIST('');
  5.03          PUT LIST('');
 10.            DECLARE  @bal1 FILE INPUT ;
 15.            DECLARE cash(5) DEC(6);
 20.            DECLARE inv(5) DEC(6);
 25.            DECLARE ar(5) DEC(6);
 30.            DECLARE ocaset(5) DEC(6);
 40.            DECLARE fasset(5) DEC(6);
 50.            DECLARE ap(5) DEC(6);
 55.            DECLARE ocliab(5) DEC(6);
 65.            DECLARE ltlib(5) DEC(6);
 70.            DECLARE cstk(5) DEC(6);
 75.            DECLARE pfd(5) DEC(6);
 80.            DECLARE retain(5) DEC(6);
 90.            DECLARE sales(5) DEC(6);
 92.            DECLARE cgs(5) DEC(6);
100.            DECLARE tox(5) DEC(6);
110.            DECLARE intexp(5) DEC(6);
120.            DECLARE taxes(5) DEC(6);
141.            DECLARE it(5);
142.            it=0;
145.    image1: IMAGE;
160.    image2: IMAGE;
165.    image3: IMAGE;
166.    image4: IMAGE;
167.            DECLARE J(10),n(10),w(10);
168.            J=0;
168.1           n=0;
168.2           w=0;
175.    image8: IMAGE;
180.            OPEN FILE(@bal1) INPUT ;
185.            ;
190.            READ FILE(@bal1) INTO(yrs) ;
195.            READ FILE(@bal1) INTO(cash) ;
200.            READ FILE(@bal1) INTO(inv) ;
205.            READ FILE(@bal1) INTO(ar) ;
210.            READ FILE(@bal1) INTO(ocaset) ;
220.            READ FILE(@bal1) INTO(fasset) ;
230.            READ FILE(@bal1) INTO(ap) ;
235.            READ FILE(@bal1) INTO(ocliab) ;
245.            READ FILE(@bal1) INTO(ltlib) ;
250.            READ FILE(@bal1) INTO(cstk) ;
255.            READ FILE(@bal1) INTO(pfd) ;
260.            READ FILE(@bal1) INTO(retain) ;
270.            READ FILE(@bal1) INTO(sales) ;
275.            READ FILE(@bal1) INTO(cgs) ;
285.            READ FILE(@bal1) INTO(tox) ;
295.            READ FILE(@bal1) INTO(intexp) ;
305.            READ FILE(@bal1) INTO(taxes) ;
321.            PUT LIST('');
321.2           PUT LIST('To determine the cash flow you must enter the amount of deprication taken in the preceding years.');
321.25          PUT LIST('');
```

```
328.            PUT IMAGE('CASH',it(1),it(2),it(3),it(4),it(5))(image1);
329.            DO i=1 TO yrs;
330.            it(i)=inv(i);
331.            END ;
332.            PUT IMAGE('INVENT',it(1),it(2),it(3),it(4),it(5))(image1);
333.            DO i=1 TO yrs;
334.            it(i)=ar(i);
335.            END ;
336.            PUT IMAGE('A/R',it(1),it(2),it(3),it(4),it(5))(image1);
337.            DO i=1 TO yrs;
338.            it(i)=ocaset(i);
339.            END ;
340.            PUT IMAGE('OCASSET',it(1),it(2),it(3),it(4),it(5))(image1);
341.            DO i=1 TO yrs;
342.            it(i)=cash(i)+inv(i)+ar(i)+ocaset(i);
343.            END ;
344.            PUT IMAGE('TOTCA',it(1),it(2),it(3),it(4),it(5))(image1);
345.            DO i=1 TO yrs;
346.            it(i)=fasset(i);
347.            END ;
348.            PUT IMAGE('FASSET',it(1),it(2),it(3),it(4),it(5))(image1);
349.            DO i=1 TO yrs;
350.            it(i)=cash(i)+inv(i)+ar(i)+ocaset(i)+fasset(i);
351.            END ;
352.            PUT IMAGE('TOTASS',it(1),it(2),it(3),it(4),it(5))(image1);
353.            PUT LIST('');
354.            PUT LIST('');
355.            DO i=1 TO yrs;
356.            it(i)=ap(i);
357.            END ;
358.            PUT IMAGE('A/P',it(1),it(2),it(3),it(4),it(5))(image1);
359.            DO i=1 TO yrs;
360.            it(i)=ocliab(i);
361.            END ;
362.            PUT IMAGE('OCLIAB',it(1),it(2),it(3),it(4),it(5))(image1);
363.            DO i=1 TO yrs;
364.            it(i)=ap(i)+ocliab(i);
365.            END ;
366.            PUT IMAGE('TOTCT',it(1),it(2),it(3),it(4),it(5))(image2);
367.            DO i=1 TO yrs;
368.            it(i)=ltlib(i);
369.            END ;
370.            PUT IMAGE('LTLIAB',it(1),it(2),it(3),it(4),it(5))(image1);
371.            DO i=1 TO yrs;
372.            it(i)=cstk(i);
373.            END ;
374.            PUT IMAGE('CSTK',it(1),it(2),it(3),it(4),it(5))(image1);
375.            DO i=1 TO yrs;
376.            it(i)=pfd(i);
377.            END ;
378.            PUT IMAGE('PFD',it(1),it(2),it(3),it(4),it(5))(image1);
379.            DO i=1 TO yrs;
380.            it(i)=retain(i);
381.            END ;
382.            PUT IMAGE('RETAIN',it(1),it(2),it(3),it(4),it(5))(image1);
383.            DO i=1 TO yrs;
384.            it(i)=ap(i)+ocliab(i)+ltlib(i)+cstk(i)+pfd(i)+retain(i);
```

```
 585.            END Dave2a;
?
```

```
 15.            DECLARE cash(5) DEC(6);
 20.            DECLARE inv(5) DEC(6);
 25.            DECLARE ar(5) DEC(6);
 30.            DECLARE ocaset(5) DEC(6);
 35.            DECLARE fasset(5) DEC(6);
 40.            DECLARE ap(5) DEC(6);
 45.            ;
 50.            DECLARE ocliab(5) DEC(6);
 52.            DECLARE ltlib(5) DEC(6);
 55.            DECLARE cstk(5) DEC(6);
 60.            DECLARE pfd(5) DEC(6);
 65.            DECLARE retain(5) DEC(6);
 70.            DECLARE sales(5) DEC(6);
 75.            DECLARE cgs(5) DEC(6);
 80.            DECLARE tox(5) DEC(6);
 85.            DECLARE intexp(5) DEC(6);
 90.            DECLARE taxes(5) DEC(6);
141.            DECLARE it(5);
142.            it=0;
145.    image1: IMAGE;
----------        ----------.--   ----------.--   ----------.--   ----------.--   ----------.--
160.    image2: IMAGE;
----------        ----------.--   ----------.--   ----------.--   ----------.--   ----------.--
165.    image3: IMAGE;
----------        ----------.--   ----------.--   ----------.--   ----------.--   ----------.--
175.    image8: IMAGE;
---------      -.---    -.---    -.---    -.---    -.---    -.---
180.            OPEN FILE(@bal1) INPUT ;
185.            ;
190.            READ FILE(@bal1) INTO(yrs) ;
195.            READ FILE(@bal1) INTO(cash) ;
200.            READ FILE(@bal1) INTO(inv) ;
205.            READ FILE(@bal1) INTO(ar) ;
210.            READ FILE(@bal1) INTO(ocaset) ;
220.            READ FILE(@bal1) INTO(fasset) ;
230.            READ FILE(@bal1) INTO(ap) ;
235.            READ FILE(@bal1) INTO(ocliab) ;
245.            READ FILE(@bal1) INTO(ltlib) ;
250.            READ FILE(@bal1) INTO(cstk) ;
255.            READ FILE(@bal1) INTO(pfd) ;
260.            READ FILE(@bal1) INTO(retain) ;
270.            READ FILE(@bal1) INTO(sales) ;
275.            READ FILE(@bal1) INTO(cgs) ;
285.            READ FILE(@bal1) INTO(tox) ;
295.            READ FILE(@bal1) INTO(intexp) ;
305.            READ FILE(@bal1) INTO(taxes) ;
321.            PUT LIST('');
322.            PUT LIST('');
322.5           PUT LIST('   Period=        1       2       3       4       5     AVE');
322.6           PUT LIST('');
455.            as=0;
456.            a=0;
457.            DECLARE as(10) DEC(6);
458.            DO i=1 TO yrs;
459.            as(i)=cash(i)/sales(i);
460.            END ;
461.            a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
462.            PUT IMAGE('CASH',as(1),as(2),as(3),as(4),as(5),a);
```

```
471.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
472.          PUT IMAGE('A/R',as(1),as(2),as(3),as(4),as(5),a)(image8);
473.          DO i=1 TO yrs;
474.          as(i)=ocaset(i)/sales(i);
475.          END ;
476.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
477.          PUT IMAGE('OCASSET',as(1),as(2),as(3),as(4),as(5),a)(image8);
478.          DO i=1 TO yrs;
479.          as(i)=(cash(i)+inv(i)+ar(i)+ocaset(i))/sales(i);
480.          END ;
481.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
482.          PUT IMAGE('TOTCA',as(1),as(2),as(3),as(4),as(5),a)(image8);
483.          DO i=1 TO yrs;
484.          as(i)=fasset(i)/sales(i);
485.          END ;
486.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
487.          PUT IMAGE('FASSET',as(1),as(2),as(3),as(4),as(5),a)(image8);
488.          DO i=1 TO yrs;
489.          as(i)=(cash(i)+inv(i)+ar(i)+ocaset(i)+fasset(i))/sales(i);
490.          END ;
491.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
492.          PUT IMAGE('TOTASS',as(1),as(2),as(3),as(4),as(5),a)(image8);
493.          DO i=1 TO yrs;
494.          as(i)=ap(i)/sales(i);
495.          END ;
496.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
497.          PUT IMAGE('A/P',as(1),as(2),as(3),as(4),as(5),a)(image8);
498.          DO i=1 TO yrs;
499.          as(i)=ocliab(i)/sales(i);
500.          END ;
501.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
502.          PUT IMAGE('OCLIAB',as(1),as(2),as(3),as(4),as(5),a)(image8);
503.          DO i=1 TO yrs;
504.          as(i)=(ap(i)+ocliab(i))/sales(i);
505.          END ;
506.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
507.          PUT IMAGE('TOTCT',as(1),as(2),as(3),as(4),as(5),a)(image8);
508.          DO i=1 TO yrs;
509.          as(i)=ltlib(i)/sales(i);
510.          END ;
511.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
512.          PUT IMAGE('LTLIAB',as(1),as(2),as(3),as(4),as(5),a)(image8);
513.          DO i=1 TO yrs;
514.          as(i)=cstk(i)/sales(i);
515.          END ;
516.          a=(as(1)+as(2)+as(3)+as(4)+as(4)+as(5))/yrs;
517.          PUT IMAGE('CSTK',as(1),as(2),as(3),as(4),as(5),a)(image8);
518.          DO i=1 TO yrs;
519.          as(i)=pfd(i)/sales(i);
520.          END ;
521.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
522.          PUT IMAGE('PFD',as(1),as(2),as(3),as(4),as(5),a)(image8);
523.          DO i=1 TO yrs;
524.          as(i)=retain(i)/sales(i);
525.          END ;
526.          a=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
527.          PUT IMAGE('RETAIN',as(1),as(2),as(3),as(4),as(5),a)(image8);
528.          DO i=1 TO ...;
```

```
531.            g=(as(1)+as(2)+as(3)+as(4)+as(5))/5;
532.            PUT IMAGE('T NW/L',as(1),as(2),as(3),as(4),as(5),g)(image8);
533.            PUT LIST('');
534.            PUT LIST('');
535.            PUT LIST('');
536.            PUT LIST('');
582.1           CLOSE FILE(@ball) ;
585.            END Dave2c;
?
```

```
122.            g=(J(1)+J(2)+J(3)+J(4)+J(5))/5;
123.            PUT IMAGE('PFD',J(1),J(2),J(3),J(4),J(5),g)(image1);
124.            DO i=1 TO yrs;
125.            J(i)=retain(i)/n(i);
126.            END ;
127.            g=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
128.            PUT IMAGE('RETAIN',J(1),J(2),J(3),J(4),J(5),g)(image1);
129.            DO i=1 TO yrs;
130.            J(i)=(ap(i)+ocliab(i)+ltlib(i)+cstk(i)+pfd(i)+retain(i))/n(i);
131.            END ;
132.            g=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
133.            PUT IMAGE('T NW/L',J(1),J(2),J(3),J(4),J(5),g)(image1);
134.            PUT LIST('');
135.            PUT LIST('');
136.            PUT LIST('');
137.            CLOSE FILE(@ball) ;
138.    there:  ;
139.            END Dave25;
?
```

```
  1.    Dave2b: PROCEDURE (depr);
 10.           DECLARE  @bal1 FILE INPUT ;
 15.           DECLARE cash(5) DEC(6);
 20.           DECLARE inv(5) DEC(6);
 25.           DECLARE ar(5) DEC(6);
 30.           DECLARE ocaset(5) DEC(6);
 40.           DECLARE fasset(5) DEC(6);
 50.           DECLARE ap(5) DEC(6);
 55.           DECLARE ocliab(5) DEC(6);
 65.           DECLARE ltlib(5) DEC(6);
 70.           DECLARE cstk(5) DEC(6);
 75.           DECLARE pfd(5) DEC(6);
 80.           DECLARE retain(5) DEC(6);
 90.           DECLARE sales(5) DEC(6);
 92.           DECLARE cgs(5) DEC(6);
100.           DECLARE tox(5) DEC(6);
110.           DECLARE intexp(5) DEC(6);
120.           DECLARE taxes(5) DEC(6);
141.           DECLARE it(5);
142.           it=0;
145.    image1: IMAGE;
160.    image2: IMAGE;
165.    image3: IMAGE;
166.    image4: IMAGE;
167.           DECLARE J(10),n(10),w(10);
168.           J=0;
168.1          n=0;
168.2          w=0;
175.    image8: IMAGE;
180.           OPEN FILE(@bal1) INPUT ;
185.           ;
190.           READ FILE(@bal1) INTO(yrs) ;
195.           READ FILE(@bal1) INTO(cash) ;
200.           READ FILE(@bal1) INTO(inv) ;
205.           READ FILE(@bal1) INTO(ar) ;
210.           READ FILE(@bal1) INTO(ocaset) ;
220.           READ FILE(@bal1) INTO(fasset) ;
230.           READ FILE(@bal1) INTO(ap) ;
235.           READ FILE(@bal1) INTO(ocliab) ;
245.           READ FILE(@bal1) INTO(ltlib) ;
250.           READ FILE(@bal1) INTO(cstk) ;
255.           READ FILE(@bal1) INTO(pfd) ;
260.           READ FILE(@bal1) INTO(retain) ;
270.           READ FILE(@bal1) INTO(sales) ;
275.           READ FILE(@bal1) INTO(cgs) ;
285.           READ FILE(@bal1) INTO(tox) ;
295.           READ FILE(@bal1) INTO(intexp) ;
305.           READ FILE(@bal1) INTO(taxes) ;
392.           DO i=1 TO yrs;
393.           it(i)=sales(i);
394.           END ;
395.           PUT IMAGE('SALES',it(1),it(2),it(3),it(4),it(5))(image3);
```

```
405.            it(i)=tox(i);
406.            END ;
407.            PUT IMAGE('OP EXP',it(1),it(2),it(3),it(4),it(5))(image3);
408.            DO i=1 TO yrs;
409.            it(i)=sales(i)-cgs(i)-tox(i);
410.            END ;
411.            PUT IMAGE('EBIT',it(1),it(2),it(3),it(4),it(5))(image3);
412.            DO i=1 TO yrs;
413.            it(i)=intexp(i);
414.            END ;
415.            PUT IMAGE('INT EXP',it(1),it(2),it(3),it(4),it(5))(image3);
416.            DO i=1 TO yrs;
417.            it(i)=sales(i)-cgs(i)-tox(i)-intexp(i);
418.            END ;
419.            PUT IMAGE('EBT',it(1),it(2),it(3),it(4),it(5))(image3);
420.            DO i=1 TO yrs;
421.            it(i)=taxes(i);
422.            END ;
423.            PUT IMAGE('TAXES',it(1),it(2),it(3),it(4),it(5))(image3);
424.            DO i=1 TO yrs;
425.            it(i)=sales(i)-cgs(i)-tox(i)-intexp(i)-taxes(i);
426.            END ;
427.            PUT IMAGE('NI',it(1),it(2),it(3),it(4),it(5))(image3);
450.            PUT LIST('');
451.            PUT LIST('');
452.            PUT LIST('');
453.            PUT IMAGE('NI',it(1),it(2),it(3),it(4),it(5))(image3);
454.            DO i=1 TO yrs;
455.            it(i)=depr(i);
456.            END ;
457.            PUT IMAGE('DEPR',it(1),it(2),it(3),it(4),it(5))(image3);
458.            DO i=1 TO yrs-1;
459.            J(i+1)=inv(i+1)+ar(i+1)+ocaset(i+1)-inv(i)-ar(i)-ocaset(i);
460.            n(i+1)=ap(i+1)+ocliab(i+1)-ap(i)-ocliab(i);
461.            END ;
462.            PUT IMAGE('CHANGE TOTCA',J(2),J(3),J(4),J(5))(image4);
463.            PUT IMAGE('CHANGE TOTCT',n(2),n(3),n(4),n(5))(image4);
464.            DO i=2 TO yrs;
465.            w(i)=sales(i)-cgs(i)-tox(i)-intexp(i)-taxes(i)+depr(i)-J(i)+n(i);
466.            END ;
467.            PUT IMAGE('CASH FLOW',w(2),w(3),w(4),w(5))(image4);
468.            PUT LIST('');
469.            PUT LIST('');
470.            PUT LIST('');
583.            CLOSE FILE(@ball) ;
584.            END ;
585.            END Dave2b;
?
```

```
 20.            DECLARE inv(5) DEC(6);
 25.            DECLARE ar(5) DEC(6);
 30.            DECLARE ocaset(5) DEC(6);
 35.            DECLARE fasset(5) DEC(6);
 40.            DECLARE ap(5) DEC(6);
 45.            ;
 50.            DECLARE ocliab(5) DEC(6);
 52.            DECLARE ltlib(5) DEC(6);
 55.            DECLARE cstk(5) DEC(6);
 60.            DECLARE pfd(5) DEC(6);
 65.            DECLARE retain(5) DEC(6);
 70.            DECLARE sales(5) DEC(6);
 75.            DECLARE css(5) DEC(6);
 80.            DECLARE tox(5) DEC(6);
 85.            DECLARE intexp(5) DEC(6);
 90.            DECLARE taxes(5) DEC(6);
141.            DECLARE it(5);
142.            it=0;
145.    image1: IMAGE;
---------          ----------.--   ----------.--   ----------.--   ----------.--   ----------.--
160.    image2: IMAGE;
---------          ----------.--   ----------.--   ----------.--   ----------.--   ---------.--
165.    image3: IMAGE;
---------          ----------.--   ----------.--   ----------.--   ----------.--   ----------.--
175.    image8: IMAGE;
---------      -.---    -.---    -.---    -.---    -.---    -.---
180.            OPEN FILE(@ball) INPUT ;
185.            ;
190.            READ FILE(@ball) INTO(yrs) ;
195.            READ FILE(@ball) INTO(cash) ;
200.            READ FILE(@ball) INTO(inv) ;
205.            READ FILE(@ball) INTO(ar) ;
210.            READ FILE(@ball) INTO(ocaset) ;
220.            READ FILE(@ball) INTO(fasset) ;
230.            READ FILE(@ball) INTO(ap) ;
235.            READ FILE(@ball) INTO(ocliab) ;
245.            READ FILE(@ball) INTO(ltlib) ;
250.            READ FILE(@ball) INTO(cstk) ;
255.            READ FILE(@ball) INTO(pfd) ;
260.            READ FILE(@ball) INTO(retain) ;
270.            READ FILE(@ball) INTO(sales) ;
275.            READ FILE(@ball) INTO(css) ;
285.            READ FILE(@ball) INTO(tox) ;
295.            READ FILE(@ball) INTO(intexp) ;
305.            READ FILE(@ball) INTO(taxes) ;
455.            as=0;
456.            a=0;
457.            DECLARE as(10) DEC(6);
536.5           r=0;
537.            PUT LIST('');
538.            DO i=1 TO yrs;
539.            as(i)=sales(i)/sales(i);
540.            END ;
541.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
542.            PUT IMAGE('SALES',as(1),as(2),as(3),as(4),as(5),r)(image8);
543.            DO i=1 TO yrs;
544.            as(i)=css(i)/sales(i);
```

```
554.            as(i)=tox(i)/sales(i);
555.            END ;
556.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
557.            PUT IMAGE('OP EXP',as(1),as(2),as(3),as(4),as(5),r)(image8);
558.            DO i=1 TO yrs;
559.            as(i)=(sales(i)-cgs(i)-tox(i))/sales(i);
560.            END ;
561.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
562.            PUT IMAGE('EBIT',as(1),as(2),as(3),as(4),as(5),r)(image8);
563.            DO i=1 TO yrs;
564.            as(i)=intexp(i)/sales(i);
565.            END ;
566.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
567.            PUT IMAGE('INT EXP',as(1),as(2),as(3),as(4),as(5),r)(image8);
568.            DO i=1 TO yrs;
569.            as(i)=(sales(i)-cgs(i)-tox(i)-intexp(i))/sales(i);
570.            END ;
571.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
572.            PUT IMAGE('EBT',as(1),as(2),as(3),as(4),as(5),r)(image8);
573.            DO i=1 TO yrs;
574.            as(i)=taxes(i)/sales(i);
575.            END ;
576.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
577.            PUT IMAGE('TAXES',as(1),as(2),as(3),as(4),as(5),r)(image8);
578.            DO i=1 TO yrs;
579.            as(i)=(sales(i)-cgs(i)-tox(i)-intexp(i)-taxes(i))/sales(i);
580.            END ;
581.            r=(as(1)+as(2)+as(3)+as(4)+as(5))/yrs;
582.            PUT IMAGE('NI',as(1),as(2),as(3),as(4),as(5),r)(image8);
582.1           CLOSE FILE(@ba11) ;
584.            END ;
585.            END Dave2d;
?
```

```
  3.            PUT LIST('');
  4.            PUT LIST('');
  5.            DECLARE  ans CHAR(3) VAR;
 5.5  here:     ;
  6.            PUT LIST('Do you wish to see a print-out of the comparative Balance Sheet comparing the accounts');
  7.            PUT LIST('to Total Assets and Total Net Worth+Liabilities?');
  8.            READ INTO(ans) ;
  9.            IF upcase(ans)='YES' THEN GO TO start;
 10.            IF upcase(ans)='NO' THEN GO TO there; ELSE GO TO here;
 11.    start:  ;
 12.            DECLARE  @bal1 FILE INPUT ;
 13.            DECLARE cash(5) DEC(6);
 14.            DECLARE inv(5) DEC(6);
 15.            DECLARE ar(5) DEC(6);
 16.            DECLARE ocaset(5) DEC(6);
 17.            DECLARE fasset(5) DEC(6);
 18.            DECLARE ap(5) DEC(6);
 19.            DECLARE ocliab(5) DEC(6);
 20.            DECLARE ltlib(5) DEC(6);
 21.            DECLARE cstk(5) DEC(6);
 22.            DECLARE pfd(5) DEC(6);
 23.            DECLARE retain(5) DEC(6);
 24.            DECLARE sales(5) DEC(6);
 25.            DECLARE cgs(5) DEC(6);
 26.            DECLARE tox(5) DEC(6);
 27.            DECLARE intexp(5) DEC(6);
 28.            DECLARE taxes(5) DEC(6);
 29.            DECLARE J(10);
 30.            DECLARE n(10);
 31.            DECLARE y(10);
 32.    image1: IMAGE;
---------     -.---    -.---    -.---    -.---    -.---    -.---
 33.            OPEN FILE(@bal1) INPUT ;
 34.            READ FILE(@bal1) INTO(yrs) ;
 35.            READ FILE(@bal1) INTO(cash) ;
 36.            READ FILE(@bal1) INTO(inv) ;
 37.            READ FILE(@bal1) INTO(ar) ;
 38.            READ FILE(@bal1) INTO(ocaset) ;
 39.            READ FILE(@bal1) INTO(fasset) ;
 40.            READ FILE(@bal1) INTO(ap) ;
 41.            READ FILE(@bal1) INTO(ocliab) ;
 42.            READ FILE(@bal1) INTO(ltlib) ;
 43.            READ FILE(@bal1) INTO(cstk) ;
 44.            READ FILE(@bal1) INTO(pfd) ;
 45.            READ FILE(@bal1) INTO(retain) ;
 46.            READ FILE(@bal1) INTO(sales) ;
 47.            READ FILE(@bal1) INTO(cgs) ;
 48.            READ FILE(@bal1) INTO(tox) ;
 49.            READ FILE(@bal1) INTO(intexp) ;
 50.            READ FILE(@bal1) INTO(taxes) ;
 51.            PUT LIST('');
 52.            PUT LIST('');
 53.            PUT LIST('   Periods=       1       2       3       4       5       AVE');
 54.            PUT LIST('');
 55.            DO i=1 TO yrs;
 56.            y(i)=cash(i)+inv(i)+ar(i)+ocaset(i)+fasset(i);
 57.            n(i)=ap(i)+ocliab(i)+ltlib(i)+cstk(i)+pfd(i)+retain(i);
 58.            END ;
```

```
65.           J(i)=inv(i)/y(i);
66.           END ;
67.           a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
68.           PUT IMAGE('INVENT',J(1),J(2),J(3),J(4),J(5),a)(image1);
69.           DO i=1 TO yrs;
70.           J(i)=ar(i)/y(i);
71.           END ;
72.           a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
73.           PUT IMAGE('A/R',J(1),J(2),J(3),J(4),J(5),a)(image1);
74.           DO i=1 TO yrs;
75.           J(i)=ocaset(i)/y(i);
76.           END ;
77.           a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
78.           PUT IMAGE('OCASSET',J(1),J(2),J(3),J(4),J(5),a)(image1);
79.           DO i=1 TO yrs;
80.           J(i)=(cash(i)+inv(i)+ar(i)+ocaset(i))/y(i);
81.           END ;
82.           a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
83.           PUT IMAGE('TOTCA',J(1),J(2),J(3),J(4),J(5),a)(image1);
84.           DO i=1 TO yrs;
85.           J(i)=fasset(i)/y(i);
86.           END ;
87.           a=(J(i)+J(2)+J(3)+J(4)+J(5))/yrs;
88.           PUT IMAGE('FASSET',J(1),J(2),J(3),J(4),J(5),a)(image1);
89.           DO i=1 TO yrs;
90.           J(i)=(cash(i)+inv(i)+ar(i)+ocaset(i)+fasset(i))/y(i);
91.           END ;
92.           a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
93.           PUT IMAGE('TOTASS',J(1),J(2),J(3),J(4),J(5),a)(image1);
93.1          PUT LIST('');
93.2          PUT LIST('');
94.           DO i=1 TO yrs;
95.           J(i)=ap(i)/n(i);
96.           END ;
97.           a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
98.           PUT IMAGE('A/P',J(1),J(2),J(3),J(4),J(5),a)(image1);
99.           DO i=1 TO yrs;
100.          J(i)=ocliab(i)/n(i);
101.          END ;
102.          a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
103.          PUT IMAGE('OCLIAB',J(1),J(2),J(3),J(4),J(5),a)(image1);
104.          DO i=1 TO yrs;
105.          J(i)=(ap(i)+ocliab(i))/n(i);
106.          END ;
107.          a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
108.          PUT IMAGE('TOTCT',J(1),J(2),J(3),J(4),J(5),a)(image1);
109.          DO i=1 TO yrs;
110.          J(i)=ltlib(i)/n(i);
111.          END ;
112.          a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
113.          PUT IMAGE('LTLIAB',J(1),J(2),J(3),J(4),J(5),a)(image1);
114.          DO i=1 TO yrs;
115.          J(i)=cstk(i)/n(i);
116.          END ;
117.          a=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
118.          PUT IMAGE('CSTK',J(1),J(2),J(3),J(4),J(5),a)(image1);
119.          DO i=1 TO yrs;
120.          J(i)=...
```

```
128.              PUT IMAGE('RETAIN',J(1),J(2),J(3),J(4),J(5),g)(image1);
129.              DO i=1 TO yrs;
130.              J(i)=(ap(i)+ocliab(i)+ltlib(i)+cstk(i)+rfd(i)+retain(i))/n(i);
131.              END ;
132.              g=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
133.              PUT IMAGE('T NW/L',J(1),J(2),J(3),J(4),J(5),g)(image1);
134.              PUT LIST('');
135.              PUT LIST('');
136.              PUT LIST('');
137.              CLOSE FILE(@ball) ;
138.     there:   ;
139.              END Dave25;
?
```

```
 5.01           PUT LIST('');
 5.02           PUT LIST('');
 5.03           PUT LIST('');
 5.5            DECLARE  ans CHAR(3) VAR;
 5.55           PUT LIST('Do you wish to see a printout of the ratio analysis.');
 5.7            READ INTO(ans) ;
 5.75           IF upcase(ans)='YES' THEN GO TO start;
 5.76           IF upcase(ans)='NO' THEN GO TO here;
 5.8    start:  ;
 10.            DECLARE  @bal1 FILE INPUT ;
 15.            DECLARE cash(5) DEC(6);
 16.            DECLARE inv(5) DEC(6);
 17.            DECLARE ar(5) DEC(6);
 18.            DECLARE ocaset(5) DEC(6);
 19.            DECLARE fasset(5) DEC(6);
 20.            DECLARE ap(5) DEC(6);
 21.            DECLARE ocliab(5) DEC(6);
 22.            DECLARE ltlib(5) DEC(6);
 23.            DECLARE cstk(5) DEC(6);
 24.            DECLARE pfd(5) DEC(6);
 25.            DECLARE retain(5) DEC(6);
 26.            DECLARE sales(5) DEC(6);
 27.            DECLARE cgs(5) DEC(6);
 28.            DECLARE tox(5) DEC(6);
 29.            DECLARE intexp(5) DEC(6);
 30.            DECLARE taxes(5) DEC(6);
155.    image5: IMAGE;
-------------------    ---.---    ---.---    ---.---    ---.---    ---.---    ---.---    ---.---    ---.--- TIMES
160.    image6: IMAGE;
-------------------    --.---    --.---    --.---    --.---    --.---
165.    image7: IMAGE;
-------------------    ---.---    ---.---    ---.---    ---.---    ---.---    ---.---    ---.---    ---.--- DAYS
170.    image8: IMAGE;
-------------------    -------.--    -------.--    -------.--    -------.--    -------.--
175.           DECLARE ratio(10) DEC(6);
180.           OPEN FILE(@bal1) INPUT ;
185.           READ FILE(@bal1) INTO(yrs) ;
190.           READ FILE(@bal1) INTO(cash) ;
195.           READ FILE(@bal1) INTO(inv) ;
200.           READ FILE(@bal1) INTO(ar) ;
205.           READ FILE(@bal1) INTO(ocaset) ;
215.           READ FILE(@bal1) INTO(fasset) ;
225.           READ FILE(@bal1) INTO(ap) ;
230.           READ FILE(@bal1) INTO(ocliab) ;
240.           READ FILE(@bal1) INTO(ltlib) ;
250.           READ FILE(@bal1) INTO(cstk) ;
255.           READ FILE(@bal1) INTO(pfd) ;
260.           READ FILE(@bal1) INTO(retain) ;
270.           READ FILE(@bal1) INTO(sales) ;
275.           READ FILE(@bal1) INTO(cgs) ;
285.           READ FILE(@bal1) INTO(tox) ;
295.           READ FILE(@bal1) INTO(intexp) ;
305.           READ FILE(@bal1) INTO(taxes) ;
305.03         ;
305.06         DECLARE totca(10) DEC(6);
305.09         DECLARE totass(10) DEC(6);
305.12         DECLARE totct(10) DEC(6);
```

```
305.39          tnl(i)=cstk(i)+pfd(i)+retain(i);
305.42          margin(i)=sales(i)-cgs(i);
305.45          ebit(i)=sales(i)-cgs(i)-tox(i);
305.48          ebt(i)=sales(i)-cgs(i)-tox(i)-intexp(i);
305.51          ny(i)=sales(i)-cgs(i)-tox(i)-intexp(i)-taxes(i);
305.54          END ;
340.    cat:    PUT LIST('');
345.            PUT LIST('          Periods=          1          2          3          4          5');
350.            ratio=0;
355.            c=yrs;
356.            PUT LIST('');
357.            PUT LIST('    LIQUIDITY RATIO');
359.            PUT LIST('');
360.            DO i=1 TO c;
365.            ratio(i)=totca(i)/totct(i);
370.            END ;
375.            PUT IMAGE('Current-Ratio',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image5);
380.            DO i=1 TO c;
385.            ratio(i)=(cash(i)+ar(i)+ocaset(i))/totct(i);
390.            END ;
395.            PUT IMAGE('Quick-Ratio',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image5);
396.            DO i=1 TO c;
397.            ratio(i)=inv(i)/(cash(i)+inv(i)+ar(i)+ocaset(i)-ap(i)-ocliab(i));
398.            END ;
399.            PUT IMAGE('INV to W.C.',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
399.1           PUT LIST('');
399.2           PUT LIST('    LEVERAGE RATIO');
399.3           PUT LIST('');
400.            DO i=1 TO c;
405.            ratio(i)=(totct(i)+ltlib(i))/totass(i);
410.            END ;
415.            PUT IMAGE('Debt to Assets',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
420.            DO i=1 TO c;
425.            ratio(i)=ebit(i)/intexp(i);
430.            END ;
435.            PUT IMAGE('Intexp to EBIT',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image5);
436.            PUT LIST('');
436.5           PUT LIST('');
437.            PUT LIST('    ACTIVITY RATIO');
438.            PUT LIST('');
440.            DO i=1 TO c;
445.            ratio(i)=ar(i)/(sales(i)/365);
450.            END ;
455.            PUT IMAGE('AR/Daily Sales',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image7);
455.1           DO i=1 TO c;
455.2           ratio(i)=totct(i)/tnl(i);
455.3           END ;
455.4           PUT IMAGE('C. Liab to Net Worth',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
455.5           DO i=1 TO c;
455.6           ratio(i)=fasset(i)/tnl(i);
455.7           END ;
455.8           PUT IMAGE('F.A. to Net Worth',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
460.            DO i=1 TO c;
465.            ratio(i)=sales(i)/fasset(i);
470.            END ;
475.            PUT IMAGE('F.A. Turnover',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image5);
480.            DO i=1 TO c;
```

```
495.7           END ;
495.8           PUT IMAGE('INV Turnover',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image5);
495.81          PUT LIST('');
495.85          PUT LIST('');
495.86          PUT LIST('    PROFITABILITY RATIOS');
495.87          PUT LIST('');
500.            DO i=1 TO c;
505.            ratio(i)=ny(i)/sales(i);
510.            END ;
515.            PUT IMAGE('Profit Margin',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
520.            DO i=1 TO c;
525.            ratio(i)=tox(i)/sales(i);
530.            END ;
535.            PUT IMAGE('Operating Exp/Sales',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
540.            DO i=1 TO c;
545.            ratio(i)=ny(i)/totass(i);
550.            END ;
555.            PUT IMAGE('Return on Assets',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
560.            DO i=1 TO c;
565.            ratio(i)=ny(i)/tnl(i);
570.            END ;
575.            PUT IMAGE('Net Worth Return',ratio(1),ratio(2),ratio(3),ratio(4),ratio(5))(image6);
580.            PUT LIST('');
581.            PUT LIST('');
582.            PUT LIST('');
583.            PUT LIST('');
584.            PUT LIST('');
700.            CLOSE FILE(@bal1) ;
705.    here:   ;
710.            END Dave3;
```

```
  5.02          PUT LIST('');
  6.            DECLARE  ans CHAR(3) VAR;
  6.1           PUT LIST('Do you wish to see a sources and uses statement.');
  6.2           READ INTO(ans) ;
  6.3           IF upcase(ans)='YES' THEN GO TO mon;
  6.4           IF upcase(ans)='NO' THEN GO TO sun;
  6.5           PUT LIST('');
  6.6           PUT LIST('');
  7.    mon:    ;
 10.            DECLARE  acct CHAR(7) VAR;
 11.            DECLARE  @bal1 FILE INPUT ;
 15.            DECLARE cash(5) DEC(6);
 16.            DECLARE inv(5) DEC(6);
 17.            DECLARE ar(5) DEC(6);
 18.            DECLARE ocaset(5) DEC(6);
 19.            DECLARE fasset(5) DEC(6);
 20.            DECLARE ap(5) DEC(6);
 21.            DECLARE ocliab(5) DEC(6);
 22.            DECLARE ltlib(5) DEC(6);
 23.            DECLARE cstk(5) DEC(6);
 24.            DECLARE pfd(5) DEC(6);
 25.            DECLARE retain(5) DEC(6);
 26.            DECLARE sales(5) DEC(6);
 27.            DECLARE cgs(5) DEC(6);
 28.            DECLARE tox(5) DEC(6);
 29.            DECLARE intexp(5) DEC(6);
 30.            DECLARE taxes(5) DEC(6);
155.    image8: IMAGE;
------- ----------.--  ----------.--
160.            OPEN FILE(@bal1) INPUT ;
165.            READ FILE(@bal1) INTO(yrs) ;
170.            READ FILE(@bal1) INTO(cash) ;
175.            READ FILE(@bal1) INTO(inv) ;
180.            READ FILE(@bal1) INTO(ar) ;
185.            READ FILE(@bal1) INTO(ocaset) ;
195.            READ FILE(@bal1) INTO(fasset) ;
205.            READ FILE(@bal1) INTO(ap) ;
210.            READ FILE(@bal1) INTO(ocliab) ;
220.            READ FILE(@bal1) INTO(ltlib) ;
230.            READ FILE(@bal1) INTO(cstk) ;
235.            READ FILE(@bal1) INTO(pfd) ;
240.            READ FILE(@bal1) INTO(retain) ;
250.            READ FILE(@bal1) INTO(sales) ;
255.            READ FILE(@bal1) INTO(cgs) ;
265.            READ FILE(@bal1) INTO(tox) ;
275.            READ FILE(@bal1) INTO(intexp) ;
285.            READ FILE(@bal1) INTO(taxes) ;
305.            R=1;
310.            S=yrs;
315.            a=0;
320.    d200:   PROCEDURE (a,d);
325.            IF a=9 THEN GO TO d201;
330.            b=0;
335.            c=0;
340.            a=9;
345.    d201:   IF a=0 THEN GO TO d203;
350.            IF a<0 THEN GO TO d202;
355.            c=c+a;
```

```
405.            PUT IMAGE('TOTAL',abs(c),abs(b))(image8);
406.            c=0;
407.            b=0;
408.            a=0;
410.            PUT LIST('');
411.            PUT LIST('');
412.            PUT LIST('');
415.    d207:   END d200;
420.            DO i=R TO S-1;
425.            PUT LIST('                SOURCES          USES');
430.            PUT LIST('  YEAR',i,'to',i+1);
435.            a=cash(i)-cash(i+1);
440.            acct='CASH';
445.            CALL d200(a,acct);
450.            a=inv(i)-inv(i+1);
455.            acct='INVENT';
460.            CALL d200(a,acct);
465.            a=ar(i)-ar(i+1);
470.            acct='A/R';
475.            CALL d200(a,acct);
480.            a=ocaset(i)-ocaset(i+1);
485.            acct='OCASSET';
490.            CALL d200(a,acct);
495.            a=fasset(i)-fasset(i+1);
500.            acct='FASSET';
505.            CALL d200(a,acct);
510.            a=ap(i+1)-ap(i);
515.            acct='A/P';
520.            CALL d200(a,acct);
525.            a=ocliab(i+1)-ocliab(i);
530.            acct='OCLIAB';
535.            CALL d200(a,acct);
540.            a=ltlib(i+1)-ltlib(i);
545.            acct='LTLIAB';
550.            CALL d200(a,acct);
555.            a=cstk(i+1)-cstk(i);
560.            acct='CSTK';
565.            CALL d200(a,acct);
570.            a=pfd(i+1)-pfd(i);
575.            acct='PFD';
580.            CALL d200(a,acct);
585.            a=retain(i+1)-retain(i);
590.            acct='RETAIN';
595.            CALL d200(a,acct);
600.            PUT LIST('');
605.            END ;
606.    sun:    ;
607.            PUT LIST('');
608.            PUT LIST('');
609.            PUT LIST('');
610.            END Dave4;
?
```

```
 load(dave895
?list
   5.01          PUT LIST('');
   5.02          PUT LIST('');
   6.            DECLARE  ans CHAR(3) VAR;
  10.            DECLARE  @ball FILE INPUT ;
  15.            DECLARE cash(5) DEC(6);
  16.            DECLARE inv(5) DEC(6);
  17.            DECLARE ar(5) DEC(6);
  18.            DECLARE ocaset(5) DEC(6);
  19.            DECLARE fasset(5) DEC(6);
  20.            DECLARE ap(5) DEC(6);
  21.            DECLARE ocliab(5) DEC(6);
  22.            DECLARE ltlib(5) DEC(6);
  23.            DECLARE cstk(5) DEC(6);
  24.            DECLARE pfd(5) DEC(6);
  25.            DECLARE retain(5) DEC(6);
  26.            DECLARE sales(5) DEC(6);
  27.            DECLARE cgs(5) DEC(6);
  28.            DECLARE tox(5) DEC(6);
  29.            DECLARE intexp(5) DEC(6);
  30.            DECLARE taxes(5) DEC(6);
 157.     image9: IMAGE;
--------  ----------.--  ----------.--  ---------.--  ----------.--  ---------.--
 163.            DECLARE f(5);
 164.            DECLARE w(5);
 165.            DECLARE r(5);
 167.            DECLARE J(5);
 180.            OPEN FILE(@ball) INPUT ;
 185.            ;
 190.            READ FILE(@ball) INTO(yrs) ;
 195.            READ FILE(@ball) INTO(cash) ;
 200.            READ FILE(@ball) INTO(inv) ;
 205.            READ FILE(@ball) INTO(ar) ;
 210.            READ FILE(@ball) INTO(ocaset) ;
 220.            READ FILE(@ball) INTO(fasset) ;
 230.            READ FILE(@ball) INTO(ap) ;
 235.            READ FILE(@ball) INTO(ocliab) ;
 245.            READ FILE(@ball) INTO(ltlib) ;
 250.            READ FILE(@ball) INTO(cstk) ;
 255.            READ FILE(@ball) INTO(pfd) ;
 260.            READ FILE(@ball) INTO(retain) ;
 270.            READ FILE(@ball) INTO(sales) ;
 275.            READ FILE(@ball) INTO(cgs) ;
 285.            READ FILE(@ball) INTO(tox) ;
 295.            READ FILE(@ball) INTO(intexp) ;
 305.            READ FILE(@ball) INTO(taxes) ;
 310.5           f=0;
 311.            J=0;
 312.            r=0;
 312.5           h=0;
 312.7           g=0;
 313.            w=0;
 313.5           PUT LIST('');
 314.            yrspro=5;
 314.1           PUT LIST('');
 314.2           PUT LIST('');
```

```
            q=(f(1)+f(2)+f(3)+f(4)+f(5))*yrs;
            x=J(1)+J(2)+J(3)+J(4)+J(5);
            DO i=1 TO yrs;
            h(i)=sales(i);
            END ;
            y=h(1)+h(2)+h(3)+h(4)+h(5);
            DO i=1 TO yrs;
            r(i)=J(i)**2;
            END ;
            p=r(1)+r(2)+r(3)+r(4)+r(5);
            s=(J(1)+J(2)+J(3)+J(4)+J(5))**2;
            b1=(q-x*y)/(yrs*p-s);
            s=(h(1)+h(2)+h(3)+h(4)+h(5))/yrs;
            m=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
            bo=s-b1*m;
            DO i=1 TO yrspro;
            w(i)=bo+b1*(i+yrs);
            END ;
            PUT LIST('Are the Sales projections printed below from a regression of your past sales a accurate');
            PUT LIST('forecast of future sales.  If they are not the computer will ask for your own projections.');
            PUT LIST('');
            PUT LIST('');
            PUT LIST('');
            PUT IMAGE('Salpro',w(1),w(2),w(3),w(4),w(5))(image9);
            READ INTO(ans) ;
            IF upcase(ans)='YES' THEN GO TO xxx;
            IF upcase(ans)='NO' THEN GO TO yyy; ELSE GO TO aaa;
yyy:        w=0;
            DECLARE salpro(10) DEC(6);
            DO i=1 TO yrspro;
            GET LIST(salpro(i));
            w(i)=salpro(i);
            END ;
xxx:        PUT LIST('');
            PUT LIST('');
            PUT LIST('');
            PUT LIST('Balance Sheet Projections based on Sales relationship for Years');
            PUT LIST('                    1              2              3              4              5');
            PUT LIST('');
            DO i=1 TO yrs;
            r(i)=cash(i)/sales(i);
            END ;
            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
            DO i=1 TO yrspro;
            f(i)=t*w(i);
            END ;
            PUT IMAGE('CASH',f(1),f(2),f(3),f(4),f(5))(image9);
            DO i=1 TO yrs;
            r(i)=inv(i)/sales(i);
            END ;
            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
            DO i=1 TO yrspro;
            f(i)=t*w(i);
            END ;
            PUT IMAGE('INVENY',f(1),f(2),f(3),f(4),f(5))(image9);
            DO i=1 TO yrs;
            r(i)=ar(i)/sales(i);
            END ;
```

```
369.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
370.            DO i=1 TO yrspro;
371.            f(i)=t*w(i);
372.            END ;
373.            PUT IMAGE('OCASSET',f(1),f(2),f(3),f(4),f(5))(image9);
374.            DO i=1 TO yrs;
375.            r(i)=(cash(i)+inv(i)+ar(i)+ocaset(i))/sales(i);
376.            END ;
377.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
378.            DO i=1 TO yrspro;
379.            f(i)=t*w(i);
380.            END ;
381.            PUT IMAGE('TOTCA',f(1),f(2),f(3),f(4),f(5))(image9);
382.            DO i=1 TO yrs;
383.            r(i)=fasset(i)/sales(i);
384.            END ;
385.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
386.            DO i=1 TO yrspro;
387.            f(i)=t*w(i);
388.            END ;
389.            PUT IMAGE('FASSET',f(1),f(2),f(3),f(4),f(5))(image9);
390.            DO i=1 TO yrs;
391.            r(i)=(cash(i)+inv(i)+ar(i)+ocaset(i)+fasset(i))/sales(i);
392.            END ;
393.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
394.            DO i=1 TO yrspro;
395.            f(i)=t*w(i);
396.            END ;
397.            PUT IMAGE('TOTASS',f(1),f(2),f(3),f(4),f(5))(image9);
398.            PUT LIST('');
399.            PUT LIST('');
402.            DO i=1 TO yrs;
403.            r(i)=ap(i)/sales(i);
404.            END ;
405.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
406.            DO i=1 TO yrspro;
407.            f(i)=t*w(i);
408.            END ;
409.            PUT IMAGE('A/P',f(1),f(2),f(3),f(4),f(5))(image9);
410.            DO i=1 TO yrs;
411.            r(i)=ocliab(i)/sales(i);
412.            END ;
413.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
414.            DO i=1 TO yrspro;
415.            f(i)=t*w(i);
416.            END ;
417.            PUT IMAGE('OCLIAB',f(1),f(2),f(3),f(4),f(5))(image9);
418.            DO i=1 TO yrs;
419.            r(i)=(ap(i)+ocliab(i))/sales(i);
420.            END ;
421.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
422.            DO i=1 TO yrspro;
423.            f(i)=t*w(i);
424.            END ;
425.            PUT IMAGE('TOTCT',f(1),f(2),f(3),f(4),f(5))(image9);
426.            DO i=1 TO yrs;
427.            r(i)=ltlib(i)/sales(i);
```

```
437.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
438.            DO i=1 TO yrspro;
439.            f(i)=t*w(i);
440.            END ;
441.            PUT IMAGE('CSTK',f(1),f(2),f(3),f(4),f(5))(image9);
442.            DO i=1 TO yrs;
443.            r(i)=pfd(i)/sales(i);
444.            END ;
445.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
446.            DO i=1 TO yrspro;
447.            f(i)=t*w(i);
448.            END ;
449.            PUT IMAGE('PFD',f(1),f(2),f(3),f(4),f(5))(image9);
450.            DO i=1 TO yrs;
451.            r(i)=retain(i)/sales(i);
452.            END ;
453.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
454.            DO i=1 TO yrspro;
455.            f(i)=t*w(i);
456.            END ;
457.            PUT IMAGE('RETAIN',f(1),f(2),f(3),f(4),f(5))(image9);
458.            DO i=1 TO yrs;
459.            r(i)=(ap(i)+ocliab(i)+ltlib(i)+cstk(i)+pfd(i)+retain(i))/sales(i);
460.            END ;
461.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
462.            DO i=1 TO yrspro;
463.            f(i)=t*w(i);
464.            END ;
465.            PUT IMAGE('T NW/L',f(1),f(2),f(3),f(4),f(5))(image9);
466.            PUT LIST('');
467.            PUT LIST('');
468.    there:  ;
490.            END Dave85;
?
```

```
 10.             DECLARE  @ball FILE INPUT ;
 15.             DECLARE cash(5) DEC(6);
 16.             DECLARE inv(5) DEC(6);
 17.             DECLARE ar(5) DEC(6);
 18.             DECLARE ocaset(5) DEC(6);
 19.             DECLARE fasset(5) DEC(6);
 20.             DECLARE ap(5) DEC(6);
 21.             DECLARE ocliab(5) DEC(6);
 22.             DECLARE ltlib(5) DEC(6);
 23.             DECLARE cstk(5) DEC(6);
 24.             DECLARE pfd(5) DEC(6);
 25.             DECLARE retain(5) DEC(6);
 26.             DECLARE sales(5) DEC(6);
 27.             DECLARE cgs(5) DEC(6);
 28.             DECLARE tox(5) DEC(6);
 29.             DECLARE intexp(5) DEC(6);
 30.             DECLARE taxes(5) DEC(6);
157.    image9: IMAGE;
158.    image2: IMAGE;
163.             DECLARE f(5);
164.             DECLARE w(5);
165.             DECLARE r(5);
167.             DECLARE J(5);
180.             OPEN FILE(@ball) INPUT ;
185.             ;
190.             READ FILE(@ball) INTO(yrs) ;
195.             READ FILE(@ball) INTO(cash) ;
200.             READ FILE(@ball) INTO(inv) ;
205.             READ FILE(@ball) INTO(ar) ;
210.             READ FILE(@ball) INTO(ocaset) ;
220.             READ FILE(@ball) INTO(fasset) ;
230.             READ FILE(@ball) INTO(ap) ;
235.             READ FILE(@ball) INTO(ocliab) ;
245.             READ FILE(@ball) INTO(ltlib) ;
250.             READ FILE(@ball) INTO(cstk) ;
255.             READ FILE(@ball) INTO(pfd) ;
260.             READ FILE(@ball) INTO(retain) ;
270.             READ FILE(@ball) INTO(sales) ;
275.             READ FILE(@ball) INTO(cgs) ;
285.             READ FILE(@ball) INTO(tox) ;
295.             READ FILE(@ball) INTO(intexp) ;
305.             READ FILE(@ball) INTO(taxes) ;
310.5            f=0;
311.             J=0;
312.             r=0;
312.5            n=0;
313.5            PUT LIST('');
314.1            PUT LIST('');
314.2            PUT LIST('');
317.05           PUT LIST('');
317.1            DO i=1 TO yrs;
317.2            J(i)=i+1-1;
317.3            END ;
318.             DECLARE n(5);
319.             DO i=1 TO yrs;
```

```
329.            r(i)=J(i)**2;
330.            END ;
331.            p=r(1)+r(2)+r(3)+r(4)+r(5);
332.            s=(J(1)+J(2)+J(3)+J(4)+J(5))**2;
333.            b1=(g-x*y)/(yrs*p-s);
334.            g=(n(1)+n(2)+n(3)+n(4)+n(5))/yrs;
335.            m=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
336.            bo=g-b1*m;
337.            DO i=1 TO 5;
338.            w(i)=bo+b1*(i+yrs);
339.            END ;
340.            PUT LIST('Presented below are the growth patterns of your Total Assets and Net Worth.  If they are not correct');
341.            PUT LIST('the computer will ask for your own projection.');
342.            PUT LIST('');
343.            PUT LIST('');
344.    aaa:    PUT IMAGE('TOTAL ASSET Projection',w(1),w(2),w(3),w(4),w(5))(image9);
345.            READ INTO(ans) ;
346.            IF upcase(ans)='YES' THEN GO TO xxx;
347.            IF upcase(ans)='NO' THEN GO TO yyy; ELSE GO TO aaa;
348.    yyy:    w=0;
349.            PUT LIST('');
349.5           DECLARE project(5) DEC(6);
350.            DO i=1 TO 5;
351.            GET LIST(project(i));
351.5           w(i)=project(i);
352.            END ;
353.    xxx:    PUT LIST('');
354.            PUT LIST('');
355.            PUT LIST('Balance Sheet Projections based on a Total Asset and Total NW+liability relationship for the Years');
356.            PUT LIST('                        1              2              3              4              5');
357.            PUT LIST('');
358.            J=0;
359.            DO i=1 TO yrs;
360.            J(i)=cash(i)/n(i);
361.            END ;
362.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
363.            DO i=1 TO 5;
364.            f(i)=t*w(i);
365.            END ;
366.            PUT IMAGE('CASH',f(1),f(2),f(3),f(4),f(5))(image2);
367.            DO i=1 TO yrs;
368.            J(i)=inv(i)/n(i);
369.            END ;
370.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
371.            DO i=1 TO 5;
372.            f(i)=t*w(i);
373.            END ;
374.            PUT IMAGE('INV',f(1),f(2),f(3),f(4),f(5))(image2);
375.            DO i=1 TO yrs;
376.            J(i)=ar(i)/n(i);
377.            END ;
378.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
379.            DO i=1 TO 5;
380.            f(i)=t*w(i);
381.            END ;
382.            PUT IMAGE('A/R',f(1),f(2),f(3),f(4),f(5))(image2);
383.            DO i=1 TO yrs;
384.            ...
```

```
393.            END ;
394.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
395.            DO i=1 TO 5;
396.            f(i)=t*w(i);
397.            END ;
398.            PUT IMAGE('TOTCA',f(1),f(2),f(3),f(4),f(5))(image2);
399.            DO i=1 TO yrs;
400.            J(i)=fasset(i)/n(i);
401.            END ;
402.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
403.            DO i=1 TO 5;
404.            f(i)=t*w(i);
405.            END ;
406.            PUT IMAGE('FASSET',f(1),f(2),f(3),f(4),f(5))(image2);
414.            PUT IMAGE('TOTASS',w(1),w(2),w(3),w(4),w(5))(image2);
415.            DO i=1 TO yrs;
416.            J(i)=ap(i)/n(i);
417.            END ;
418.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
419.            DO i=1 TO 5;
420.            f(i)=t*w(i);
421.            END ;
422.            PUT LIST('');
423.            PUT IMAGE('A/P',f(1),f(2),f(3),f(4),f(5))(image2);
424.            DO i=1 TO yrs;
425.            J(i)=ocliab(i)/n(i);
426.            END ;
427.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
428.            DO i=1 TO 5;
429.            f(i)=t*w(i);
430.            END ;
431.            PUT IMAGE('OCLIAB',f(1),f(2),f(3),f(4),f(5))(image2);
432.            DO i=1 TO yrs;
433.            J(i)=(ap(i)+ocliab(i))/n(i);
434.            END ;
435.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
436.            DO i=1 TO 5;
437.            f(i)=t*w(i);
438.            END ;
439.            PUT IMAGE('TOTCT',f(1),f(2),f(3),f(4),f(5))(image2);
440.            DO i=1 TO yrs;
441.            J(i)=ltlib(i)/n(i);
442.            END ;
443.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
444.            DO i=1 TO 5;
445.            f(i)=t*w(i);
446.            END ;
447.            PUT IMAGE('LTLIAB',f(1),f(2),f(3),f(4),f(5))(image2);
448.            DO i=1 TO yrs;
449.            J(i)=cstk(i)/n(i);
450.            END ;
451.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
452.            DO i=1 TO 5;
453.            f(i)=t*w(i);
454.            END ;
455.            PUT IMAGE('CSTK',f(1),f(2),f(3),f(4),f(5))(image2);
456.            DO i=1 TO yrs;
```

```
460.            DO i=1 TO 5;
461.            f(i)=t*w(i);
462.            END ;
463.            PUT IMAGE('PFD',f(1),f(2),f(3),f(4),f(5))(imase2);
464.            DO i=1 TO yrs;
465.            J(i)=retain(i)/n(i);
466.            END ;
467.            t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
468.            DO i=1 TO 5;
469.            f(i)=t*w(i);
470.            END ;
471.            PUT IMAGE('RETAIN',f(1),f(2),f(3),f(4),f(5))(imase2);
475.            PUT IMAGE('T NW/L',w(1),w(2),w(3),w(4),w(5))(imase2);
476.            PUT LIST('');
477.            PUT LIST('');
478.            END Dave87;
?
```

```
 20.             DECLARE inv(5) DEC(6);
 25.             DECLARE ar(5) DEC(6);
 30.             DECLARE ocaset(5) DEC(6);
 40.             DECLARE fasset(5) DEC(6);
 50.             DECLARE ap(5) DEC(6);
 55.             DECLARE ocliab(5) DEC(6);
 65.             DECLARE ltlib(5) DEC(6);
 70.             DECLARE cstk(5) DEC(6);
 75.             DECLARE pfd(5) DEC(6);
 80.             DECLARE retain(5) DEC(6);
 90.             DECLARE sales(5) DEC(6);
 92.             DECLARE css(5) DEC(6);
100.             DECLARE tox(5) DEC(6);
110.             DECLARE intexp(5) DEC(6);
120.             DECLARE taxes(5) DEC(6);
121.             DECLARE v(5) DEC(6);
122.             v=0;
157.     imase9: IMAGE;
-------------------------  ----------  ----------  ----------  ----------  ----------
163.             DECLARE f(5) DEC(6);
164.             DECLARE w(5) DEC(6);
165.             DECLARE r(5) DEC(6);
167.             DECLARE J(5) DEC(6);
172.             DECLARE ai(10) DEC(6);
172.2            ai=0;
173.             DECLARE pr(10) DEC(6);
173.2            pr=0;
174.             DECLARE clt(5) DEC(6);
180.             OPEN FILE(@ball) INPUT ;
185.             ;
190.             READ FILE(@ball) INTO(yrs) ;
195.             READ FILE(@ball) INTO(cash) ;
200.             READ FILE(@ball) INTO(inv) ;
205.             READ FILE(@ball) INTO(ar) ;
210.             READ FILE(@ball) INTO(ocaset) ;
220.             READ FILE(@ball) INTO(fasset) ;
230.             READ FILE(@ball) INTO(ap) ;
235.             READ FILE(@ball) INTO(ocliab) ;
245.             READ FILE(@ball) INTO(ltlib) ;
250.             READ FILE(@ball) INTO(cstk) ;
255.             READ FILE(@ball) INTO(pfd) ;
260.             READ FILE(@ball) INTO(retain) ;
270.             READ FILE(@ball) INTO(sales) ;
275.             READ FILE(@ball) INTO(css) ;
285.             READ FILE(@ball) INTO(tox) ;
295.             READ FILE(@ball) INTO(intexp) ;
300.             READ FILE(@ball) INTO(taxes) ;
310.5            f=0;
311.             J=0;
311.03  boat:    ;
311.04           PUT LIST('');
311.06           PUT LIST('Were the Sales Projections from the regression accurate?  If not the computer will ask for the correct  S
311.09           READ INTO(ans) ;
311.1            PUT LIST('');
311.11           PUT LIST('');
311.12           IF upcase(ans)='YES' THEN GO TO sar;
311.13           IF upcase(ans)='NO' THEN GO TO car; ELSE GO TO boat;
311.14           DECLARE salse('5')
```

```
315.            J(i)=i+1-1;
316.            END ;
317.            DECLARE h(5) DEC(6);
318.            DO i=1 TO yrs;
319.            f(i)=J(i)*sales(i);
320.            END ;
321.            a=(f(1)+f(2)+f(3)+f(4)+f(5))*yrs;
322.            x=J(1)+J(2)+J(3)+J(4)+J(5);
323.            DO i=1 TO yrs;
324.            h(i)=sales(i);
325.            END ;
326.            y=h(1)+h(2)+h(3)+h(4)+h(5);
327.            DO i=1 TO yrs;
328.            r(i)=J(i)**2;
329.            END ;
330.            p=r(1)+r(2)+r(3)+r(4)+r(5);
331.            s=(J(1)+J(2)+J(3)+J(4)+J(5))**2;
332.            b1=(a-x*y)/(yrs*p-s);
333.            g=(h(1)+h(2)+h(3)+h(4)+h(5))/yrs;
334.            m=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
335.            bo=g-b1*m;
336.            DO i=1 TO 5;
337.            w(i)=bo+b1*(i+yrs);
338.            END ;
468.    cat:    DO i=1 TO 5;
468.5           r(i)=w(i);
470.1           PUT LIST('');
470.2           PUT LIST('');
470.3           PUT LIST('Enter the amount of annual debt payments due of the current long term loans for the next 5 years.');
470.33          ;
470.36          DO i=1 TO 5;
470.39          GET LIST(clt(i));
470.42          END ;
470.43          PUT LIST('');
470.44          PUT LIST('');
470.45          PUT LIST('Enter the amount of new debt requested.');
470.48          GET LIST(db);
470.49          PUT LIST('');
470.5           PUT LIST('');
470.51          PUT LIST('Enter the interest rate and the term of the new loan.');
470.54          GET LIST(R);
470.57          GET LIST(pb);
470.6           DECLARE depr(5) DEC(6);
470.61          PUT LIST('');
470.65          PUT LIST('Please enter the approx. amount of deprication to be taken in the next five years.');
470.7           DO i=1 TO 5;
470.75          GET LIST(depr(i));
470.8           END ;
470.85          PUT LIST('');
470.9           PUT LIST('');
470.92          PUT LIST('Projected Income for Years');
470.93          PUT LIST('                          1          2          3          4          5 ');
470.95          PUT LIST('');
471.            PUT IMAGE('SALPRO',w(1),w(2),w(3),w(4),w(5))(image9);
471.5           r=0;
472.            DO i=1 TO yrs;
473.            r(i)=css(i)/sales(i);
```

```
483.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
484.            DO i=1 TO yrspro;
485.            f(i)=t*w(i);
486.            END ;
487.            PUT IMAGE('MARGIN',f(1),f(2),f(3),f(4),f(5))(image9);
488.            DO i=1 TO yrs;
489.            r(i)=tox(i)/sales(i);
490.            END ;
491.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
492.            DO i=1 TO yrspro;
493.            f(i)=t*w(i);
494.            END ;
495.            PUT IMAGE('OP EXP',f(1),f(2),f(3),f(4),f(5))(image9);
496.            DO i=1 TO yrs;
497.            r(i)=(sales(i)-cgs(i)-tox(i))/sales(i);
498.            END ;
499.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
500.            DO i=1 TO yrspro;
501.            f(i)=t*w(i);
502.            END ;
503.            PUT IMAGE('EBIT',f(1),f(2),f(3),f(4),f(5))(image9);
504.            i=yrs;
505.            x=intexp(i);
506.            PUT IMAGE('INTEXP',x,x,x,x,x)(image9);
512.            DO i=1 TO yrs;
513.            r(i)=(sales(i)-cgs(i)-tox(i))/sales(i);
514.            END ;
515.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
516.            DO i=1 TO yrspro;
517.            f(i)=t*w(i);
517.5           v(i)=f(i)-x;
518.            END ;
519.            PUT IMAGE('EBT',v(1),v(2),v(3),v(4),v(5))(image9);
519.1           r=0;
520.            DO i=1 TO yrs;
521.            r(i)=taxes(i)/sales(i);
522.            END ;
523.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
524.            DO i=1 TO yrspro;
525.            f(i)=t*w(i);
526.            END ;
527.            PUT IMAGE('TAXES',f(1),f(2),f(3),f(4),f(5))(image9);
528.            DO i=1 TO yrs;
529.            r(i)=(sales(i)-cgs(i)-tox(i)-taxes(i))/sales(i);
530.            END ;
531.            t=(r(1)+r(2)+r(3)+r(4)+r(5))/yrs;
532.            DO i=1 TO yrspro;
533.            f(i)=t*w(i);
533.5           v(i)=f(i)-x;
534.            END ;
535.            PUT IMAGE('NI',v(1),v(2),v(3),v(4),v(5))(image9);
535.01          PUT LIST('');
535.03          PUT LIST('');
535.1           r=0;
535.2           f=0;
535.3           h=0;
536.            PUT IMAGE('NI',v(1),v(2),v(3),v(4),v(5))(image9);
```

```
536.55          DO i=1 TO 5-1;
536.6           f(i+1)=r(i+1)-r(i);
536.65          END ;
536.7           PUT IMAGE('CHANGE TOTCA',f(1),f(2),f(3),f(4),f(5))(image9);
536.75          DO i=1 TO yrs;
536.8           J(i)=(ap(i)+ocliab(i))/sales(i);
536.85          END ;
536.9           t=(J(1)+J(2)+J(3)+J(4)+J(5))/yrs;
536.95          DO i=1 TO 5;
537.            r(i)=t*w(i);
537.05          END ;
537.1           h(1)=r(1)-ap(yrs)-ocliab(yrs);
537.15          DO i=1 TO 5-1;
537.2           h(i+1)=r(i+1)-r(i);
537.25          END ;
537.3           PUT IMAGE('CHANGE TOTCT',h(1),h(2),h(3),h(4),h(5))(image9);
537.35          DO i=1 TO 5;
537.4           w(i)=v(i)+depr(i)-f(i)+h(i);
537.45          END ;
537.5           PUT IMAGE('CASH FLOW',w(1),w(2),w(3),w(4),w(5))(image9);
537.6           PUT LIST('');
540.            PUT LIST('CASH AVA FOR DEBT');
541.            PUT IMAGE('PAYMENTS AND DIVIDENDS',w(1),w(2),w(3),w(4),w(5))(image9);
542.            PUT IMAGE('CURRENT DEBT PAYMENT',clt(1),clt(2),clt(3),clt(4),clt(5))(image9);
543.            i=pb;
544.            int=db*R*i/2;
545.            pa=(db+int)/i;
545.5           y=db;
546.            DO i=1 TO pb;
547.            ai(i)=y*R;
548.            pr(i)=pa-ai(i);
549.            y=y-pr(i);
550.            END ;
551.            PUT IMAGE('NEW DEBT PAYMENT',pa,pa,pa,pa,pa)(image9);
552.            PUT IMAGE('INTEXP ON NEW DEBT',ai(1),ai(2),ai(3),ai(4),ai(5))(image9);
553.            PUT IMAGE('PRINCIPLE ON NEW DEBT',pr(1),pr(2),pr(3),pr(4),pr(5))(image9);
554.            DO i=1 TO 5;
555.            J(i)=w(i)-clt(i)-pa+.5*ai(i);
556.            END ;
557.            PUT LIST('');
558.            PUT LIST('CASH AVAIABLE AFTER');
559.            PUT IMAGE('DEBT PAYMENTS',J(1),J(2),J(3),J(4),J(5))(image9);
560.            END ;
800.            END Daye9;
?
```

```
 4.05          PUT LIST('We will now perform a monthly Cash Budget.  You must enter certain new data, consisting of Monthly Sales
 4.1           PUT LIST('months plus the monthly sales for the last 2 months Just completed, Total Operating Expenses and Other mo
 4.11          PUT LIST('outlays.  These accounts will be defined below.');
 4.13          PUT LIST('');
 4.15          PUT LIST('Monthly Sales(monsal)consists of your projected monthly sales of the next 12 month period.');
 4.16          PUT LIST('Total Operating Expenses(tox) consists of purchases of material, salaries, labor expenses and other cash
 4.17          PUT LIST('Other Monthly Expenses(otpay) consists of taxes, dividends and repayment of current debt.');
 4.18          PUT LIST('');
 4.19          PUT LIST('');
11.05          PUT LIST('What is your interest rate on the line of credit.');
11.1           GET LIST(R);
11.15          DECLARE v(12) DEC(6);
 52.           DECLARE a(14) DEC(6);
 53.           DECLARE b(14) DEC(6);
 54.           DECLARE c(14) DEC(6);
 55.           DECLARE d(14) DEC(6);
 56.           DECLARE s(14) DEC(6);
 57.           DECLARE t(14) DEC(6);
 58.           DECLARE u(14) DEC(6);
 58.5          DECLARE z(12) DEC(6);
 58.6          DECLARE otpay(12) DEC(6);
 65.1          DECLARE monsal(14) DEC(6);
 65.2          DECLARE montox(12) DEC(6);
 65.5   image1: IMAGE;
 65.6   image2: IMAGE;
 65.7   image3: IMAGE;
 66.           PUT LIST('');
 67.           PUT LIST('');
 71.           PUT LIST('');
 72.           PUT LIST('Please enter the last 2 months sales first and then enter the projected sales for the next 12 month perio
 73.           DO i=1 TO 14;
 74.           GET LIST(monsal(i));
 74.5          a(i)=monsal(i);
 75.           END ;
 76.           PUT LIST('');
 77.           PUT LIST('Please enter the monthly Cash Operating outlays projected for the next 12 months as defined above.');
 78.           DO i=1 TO 12;
 79.           GET LIST(montox(i));
 79.5          END ;
 79.6          PUT LIST('');
 79.63         PUT LIST('Enter Other monthly Cash Payments in the month to be paid.');
 79.66         DO i=1 TO 12;
 79.69         GET LIST(otpay(i));
 79.72         z(i)=otpay(i);
 79.75         END ;
 80.           PUT LIST('');
 81.           PUT LIST('I need to know the minimun cash balance desired.');
 82.           GET LIST(mincah);
 83.           PUT LIST('');
 84.           PUT LIST('What percentage of Total Operating Expenses are allocated to material purchases.');
 85.           GET LIST(purch);
 86.           PUT LIST('');
 87.           PUT LIST('What percentage of sales are collected in first month of sales, second month and third month.');
 87.5          DECLARE colect(3);
 88.           DO i=1 TO 3;
```

```
                a=a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+a(9)+a(10)+a(11)+a(12)+a(13)+a(14);
 96.            PUT IMAGE('MONTHLY SALES',a(3),a(4),a(5),a(6),a(7),a(8),a(9),a(10),a(11),a(12),a(13),a(14),a)(image2);
 97.            PUT LIST('COLLECTIONS');
 98.            DO i=1 TO 14;
 99.            b(i)=monsal(i)*colect(1);
100.            END ;
101.            PUT IMAGE('FIRST MONTH',colect(1),b(3),b(4),b(5),b(6),b(7),b(8),b(9),b(10),b(11),b(12),b(13),b(14))(image3);
102.            DO i=1 TO 14;
103.            c(i)=monsal(i)*colect(2);
104.            END ;
105.            PUT IMAGE('SECOND MONTH',colect(2),c(2),c(3),c(4),c(5),c(6),c(7),c(8),c(9),c(10),c(11),c(12),c(13))(image3);
106.            DO i=1 TO 14;
107.            d(i)=monsal(i)*colect(3);
108.            END ;
109.            PUT IMAGE('THIRD MONTH',colect(3),d(1),d(2),d(3),d(4),d(5),d(6),d(7),d(8),d(9),d(10),d(11),d(12))(image3);
110.            f=b(3)+c(2)+d(1);
111.            g=b(4)+c(3)+d(2);
112.            h=b(5)+c(4)+d(3);
113.            e=b(6)+c(5)+d(4);
114.            j=b(7)+c(6)+d(5);
115.            k=b(8)+c(7)+d(6);
116.            l=b(9)+c(8)+d(7);
117.            m=b(10)+c(9)+d(8);
118.            n=b(11)+c(10)+d(9);
119.            o=b(12)+c(11)+d(10);
120.            p=b(13)+c(12)+d(11);
121.            q=b(14)+c(13)+d(12);
122.            r=f+g+h+e+j+k+l+m+n+o+p+q;
123.            PUT IMAGE('TOTAL',f,g,h,e,j,k,l,m,n,o,p,q,r)(image2);
123.5           PUT LIST(' ');
124.            DO i=1 TO 12;
125.            s(i)=montox(i);
126.            t(i)=s(i)*purch;
127.            u(i)=s(i)*(1-purch);
128.            END ;
129.            pu=t(1)+t(2)+t(3)+t(4)+t(5)+t(6)+t(7)+t(8)+t(9)+t(10)+t(11)+t(12);
130.            PUT IMAGE('PURCHASES',t(1),t(2),t(3),t(4),t(5),t(6),t(7),t(8),t(9),t(10),t(11),t(12),pu)(image2);
131.            wr=u(1)+u(2)+u(3)+u(4)+u(5)+u(6)+u(7)+u(8)+u(9)+u(10)+u(11)+u(12);
132.            PUT IMAGE('OP EXPENSES',u(1),u(2),u(3),u(4),u(5),u(6),u(7),u(8),u(9),u(10),u(11),u(12),wr)(image2);
132.3           pi=z(1)+z(2)+z(3)+z(4)+z(5)+z(6)+z(7)+z(8)+z(9)+z(10)+z(11)+z(12);
132.4           PUT IMAGE('OTHER PAYMENTS',z(1),z(2),z(3),z(4),z(5),z(6),z(7),z(8),z(9),z(10),z(11),z(12),pi)(image2);
133.            fg=f-s(1)-z(1);
134.            kj=g-s(2)-z(2);
135.            qp=h-s(3)-z(3);
136.            lk=e-s(4)-z(4);
137.            gh=j-s(5)-z(5);
138.            at=k-s(6)-z(6);
139.            it=l-s(7)-z(7);
140.            gt=m-s(8)-z(8);
141.            ok=n-s(9)-z(9);
142.            xx=o-s(10)-z(10);
143.            yy=p-s(11)-z(11);
144.            rr=q-s(12)-z(12);
146.            PUT IMAGE('NET CASH GAIN(LOSS)',fg,kj,qp,lk,gh,at,it,gt,ok,xx,yy,rr)(image2);
147.            aa=fg+casbal;
148.            bb=kj+aa;
149.            cc=qp+bb;
150.            dd=lk+cc;
```

```
160.            PUT IMAGE('BEGINNING BALANCE',casbal,aa,bb,cc,dd,ee,ff,gg,hh,JJ,kk,ll)(image2);
162.            PUT IMAGE('CUMMULATIVE CASH',aa,bb,cc,dd,ee,ff,gg,hh,JJ,kk,ll,mm)(image2);
163.            v(1)=aa-mincah;
164.            v(2)=bb-mincah;
165.            v(3)=cc-mincah;
166.            v(4)=dd-mincah;
167.            v(5)=ee-mincah;
168.            v(6)=ff-mincah;
169.            v(7)=gg-mincah;
170.            v(8)=hh-mincah;
171.            v(9)=JJ-mincah;
172.            v(10)=kk-mincah;
173.            v(11)=ll-mincah;
174.            v(12)=mm-mincah;
176.            PUT IMAGE('DESIRED LEVEL OF CASH',mincah,mincah,mincah,mincah,mincah,mincah,mincah,mincah,mincah,mincah,mincah,minc
176.1           PUT LIST('');
176.2           a=0;
177.05          DO i=1 TO 12;
177.1           a(i)=v(i)*R/12;
177.15          IF v(i)>0 THEN a(i)=0;
177.18          IF a(i)>0 THEN a(i)=0;
177.19          s(i)=v(i)+a(i);
177.2           END ;
177.25          PUT IMAGE('INTEXP ON LINE OF CREDIT',a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8),9,a(10),a(11),a(12))(image2);
177.9           PUT LIST('TOTAL LOANS NEEDED');
178.            PUT IMAGE('OR SURPLUS',s(1),s(2),s(3),s(4),s(5),s(6),s(7),s(8),s(9),s(10),s(11),s(12))(image2);
179.            PUT LIST('');
179.05  image4: IMAGE;
-----------------------------------------                                                                    ---------
179.2           pf=r-pu-pi-wr+a(1)+a(2)+a(3)+a(4)+a(5)+a(6)+a(7)+a(8)+a(9)+a(10)+a(11)+a(12);
179.3           PUT IMAGE('CASH FLOW FROM OPERATIONS',pf)(image4);
182.            PUT LIST('');
?
```