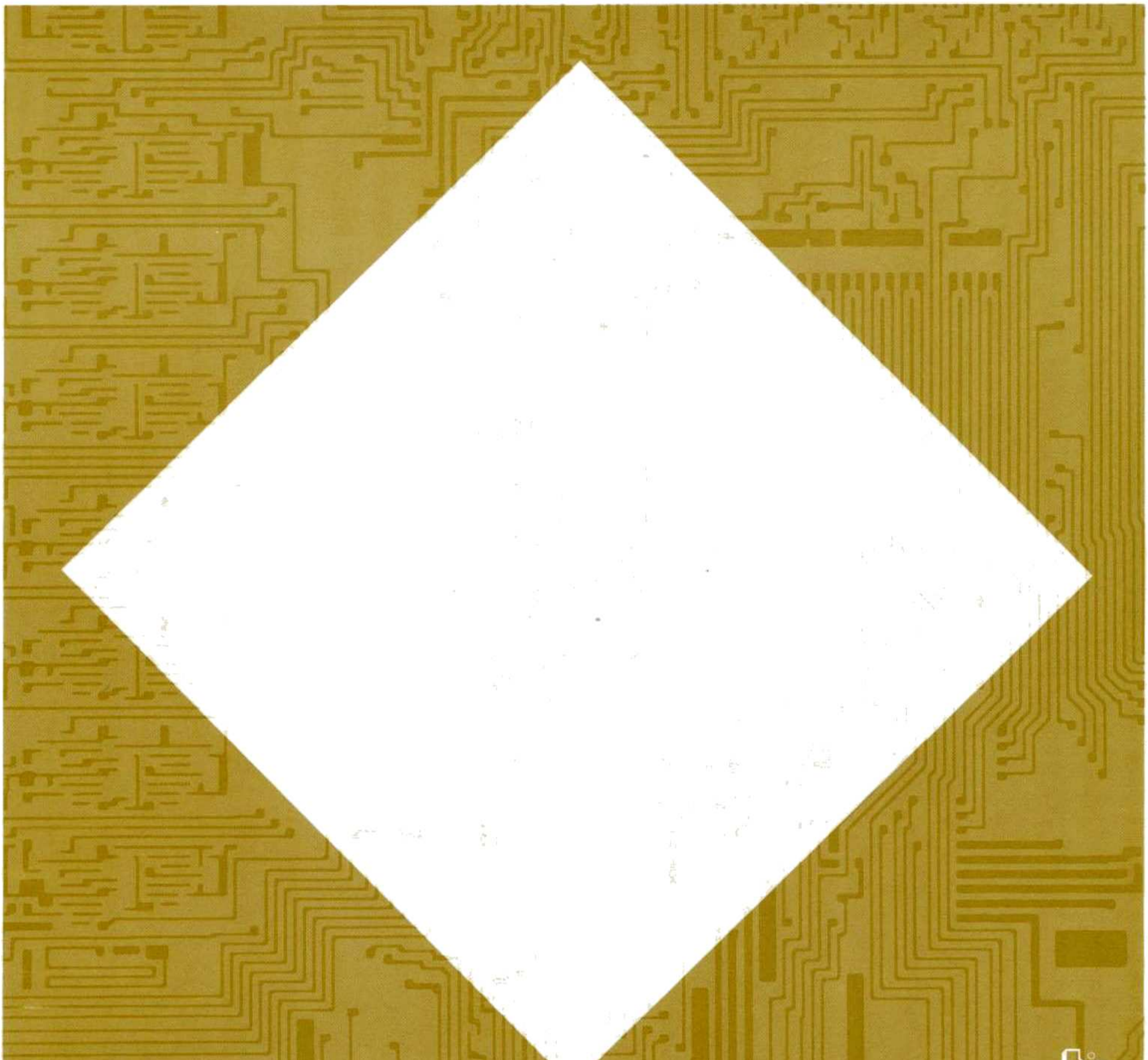
TI Programmable 58/59

Business Decisions

Using the power of your *Solid State Software*™ module



ACKNOWLEDGEMENT

Texas Instruments expresses sincere gratitude to the following for their contributions to the Business Decisions Library.

David L. Dukes
College of Business Administration
University of Missouri at St. Louis
St. Louis, Missouri

A. Dale Flowers
Department of Operations Research
Case Western Reserve University
Cleveland, Ohio

INTRODUCTION

Today's inflationary pressures and changing business conditions are placing extra demands on every businessman. Budgets, forecasts, and operating plans must be revised frequently to comprehend changes in economic and business conditions. To assist you in making timely decisions, the Business Decisions *Solid State Software** module tailors your calculator to solve a variety of problems related to business situations, without getting involved in the step-by-step mathematical computations required. With this ease of calculation, you can devote more time to estimating the values which are needed as inputs for many of the calculations. And you can determine the sensitivity of the computed values to changes, or errors, in these estimated inputs by obtaining several solutions using different input values. Before implementing any of the programs in the library, you should become familiar with its intended areas of application, and basic assumptions, if any, which may limit its use. Sound business judgment should be the overriding factor in any decision making process.

USING THIS MANUAL

Following this brief introduction, you will find the description, user instructions, and example problems for each of the 11 programs in the Business Decisions Library. Each program is easily identified by the "BD" number in the upper corner of the page. This number corresponds with the call number you use to tell the calculator which program in the *Solid State Software* module you wish to use. Principal equations used in these programs may be found in Appendix A.

The primary reference point in this manual for each program is the User Instructions. These user instructions are also available for you in the handy pocket guide furnished with the library. The program description and sample problems should be used when you first run a program, to help you understand its full capabilities and limitations. Nonmagnetic label cards to identify the user-defined keys are also included in the library. Carefully remove the cards from the sheet and insert them in the card carrying case for convenient storage. Note that a special holder has been built into the case for storage of the library module.

When using the *Solid State Software* programs as subroutines to your own programs, you will also want to check Register Contents for the program and check Program Reference Data provided in Appendix B.

USING THE OPTIONAL PRINTER

If you have the optional PC-100A printer†, a printed record of entries and results is automatic. The User Instructions and example problems are marked to show exactly which values are printed in addition to being displayed.

Use the Calculator Mounting procedure in the PC-100A Owner's Manual to mount your calculator on the PC-100A. The switch called out in Step 2 should be set to "OTHER" for your calculator. Always turn the calculator and printer off before mounting or removing the calculator.

*Trademark of Texas Instruments

†Note: The TI Programmable 58 and TI Programmable 59 will not operate on the PC-100 print cradle.

INTRODUCTION

TIPS FOR RUNNING PROGRAMS

Before you begin using the *Solid State Software* programs on your own, here are a few things to keep clearly in mind until you become familiar with your calculator.

1. Press [CLR] before running a program if you are not sure of the status of the calculator. (To be completely sure of calculator status, turn it off and on again – but remember that this will clear the program memory.)

2. Some programs will leave the calculator in fix-decimal format (see Appendix B). In that event, you should press [INV] [2nd] [fix] before running another program if this format is not desired.

3. There is no visual indication of which *Solid State Software* program has been called. If you have any doubts, the safest method is to call the desired program with [2nd] [Pgm] mm, where mm is the two-digit program number. The calculator will remain at this program number until another program is called, [RST] is pressed or the calculator is turned off.

4. A flashing display normally indicates an improper key sequence or that a numerical limit has been exceeded. When this occurs, always repeat the program sequence and check that each step is performed as directed by the User Instructions. Any unusual limits of a program are given in the User Instructions or related notes. The In Case of Difficulty portion of Appendix A in *Personal Programming* may be helpful in isolating a problem.

5. Some of the *Solid State Software* programs may run for several minutes depending on input data. If you desire to halt a running program, press the [RST] key. This is considered as an emergency halt operation which returns control to the main memory. A program must be recalled to be run again.

USING *SOLID STATE SOFTWARE* PROGRAMS AS SUBROUTINES

Any of the *Solid State Software* programs may be called as a subroutine to your own program in the main memory. Either of two program sequences may be used: 1) [2nd] [Pgm] mm (User-Defined Key) or 2) [2nd] [Pgm] mm [SBR] (Common Label). Both will send the program control to program mm, run the subroutine sequence, and then automatically return to the main program without interruption. Following [2nd] [Pgm] mm with anything other than [SBR] or a user-defined key is not a valid key sequence and can cause unwanted results.

It is very important to consider the Program Reference Data in Appendix B for any program called as a subroutine. You must plan and write your own program such that the data registers, flags, subroutine levels, parentheses levels, T-register, angular mode, etc., used by the called subroutine are allowed for in your program. In addition, a Register Contents section of each program description provides a guide to determine where data is or must be located to run the program. A sample program that calls a *Solid State Software* program as a subroutine is provided in the PROGRAMMING CONSIDERATIONS section of *Personal Programming.*

If you need to examine and study the content of a *Solid-State Software* program, you can download as described in the following paragraphs.

DOWNLOADING *SOLID STATE SOFTWARE* PROGRAMS

If you need to examine a *Solid State Software* program, it can be downloaded into the main program memory.* This will allow you to single step through a program in or out of the learn mode. It also allows using the program list or trace features of the optional printer. The only requirement for downloading a *Solid State Software* program is that the memory partition be set so there is sufficient space in the main program memory to receive the downloaded program. The key sequence to download a program is [2nd] [CP] [2nd] [Pgm] mm [2nd] [Op] 09, where mm is the program number to be downloaded. This procedure places the requested program into program memory beginning at program location 000. The downloaded program writes over any instructions previously stored in that part of program memory. Remember to press [RST] before running or tracing the downloaded program.

Please note that BD-05, BD-08, BD-10, and BD-11 cannot be downloaded in the TI Programmable 58 due to the length of these programs. Also, the memory partition must be reset from the power-up condition in the TI-58 for programs BD-02, 04, 06, 07 and 09. The key sequence to repartition the memory for BD-04 and BD-09 is 1 [2nd] [Op] 17. The sequence for BD-02, BD-06, and BD-07 is 0 [2nd] [Op] 17. Repartitioning must be performed before the downloading sequence.

The partition must be changed from the power-up condition in the TI Programmable 59 for BD-05, BD-08, BD-10, and BD-11. The key sequence to repartition the memory for BD-10 is 3 [2nd] [Op] 17. The sequence for BD-05 and BD-11 is 4 [2nd] [Op] 17. The sequence for BD-08 is 0 [2nd] [Op] 17.

REMOVING AND INSTALLING MODULES

The Business Decisions Module can easily be installed in the calculator or replaced with another. It is a good idea to leave the module in place in the calculator except when replacing it with another module. Be sure to follow these instructions when you need to remove or replace a module.

CAUTION

Be sure to touch some metal object before handling a module to prevent possible damage by static electricity.

1. **Turn the calculator OFF.** Loading or unloading the module with the calculator ON may cause the keyboard or display to lock out. Also, shorting the contacts can damage the module or calculator.

2. Slide out the small panel covering the module compartment at the bottom of the back of the calculator. (See Diagram on following page.)

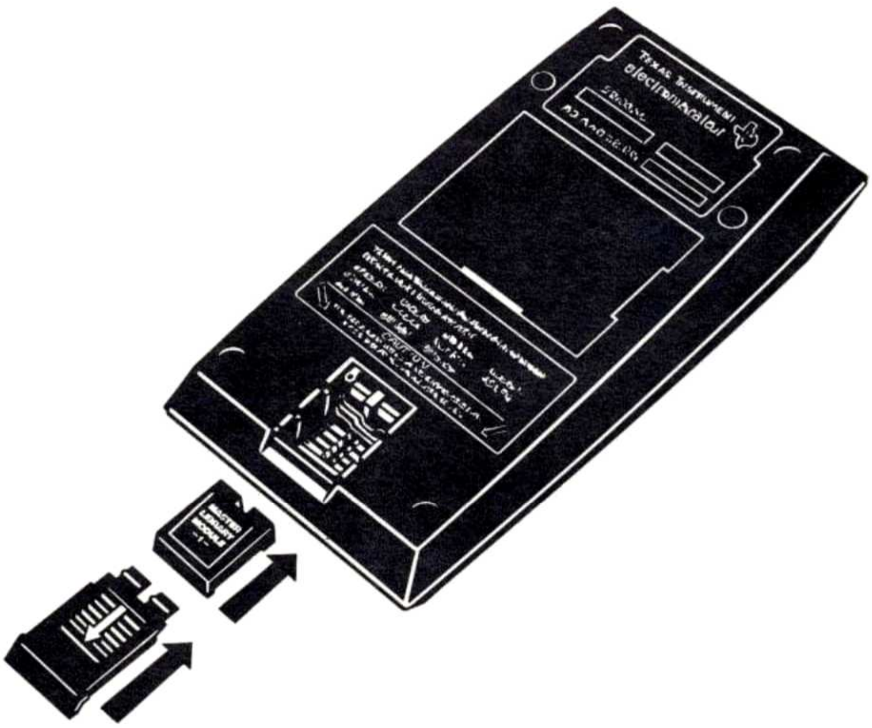
3. Remove the module. You may turn the calculator over and let the module fall out into your hand.

*Unless the library is a protected special-purpose library.

INTRODUCTION

4. Insert the module, notched end first with the labeled side up into the compartment. The module should slip into place effortlessly.

5. Replace the cover panel, securing the module against the contacts.



Don't touch the contacts inside the module compartment as damage can result.

BUSINESS DECISIONS MODULE CHECK

This program performs the following functions separately.

1. Library Module Check
2. Linear Regression Initialization

Library Module Check

When you want to know which of your *Solid State Software* modules is in the calculator without physically looking at it, you can call the Library Module check portion of the routine. If the Business Decisions Module is in the calculator, the number 9. will be displayed. This number if unique to the Business Decisions Library (other optional libraries use other identifying digits). If the calculator is attached to a PC-100A print cradle, the following will be printed:

```
BUSINESS DECISIONS
9.
```

If these results are not obtained, there is probably a malfunction in the calculator or the *Solid State Software* module. First, install the Master Library module and try running the diagnostic program, ML-01. If this program does not run properly, refer to Appendix A of *Personal Programming* for an explanation of the various procedures to be followed when you have difficulties.

Linear Regression Initialization

This routine initializes the calculator for linear regression by clearing data registers R_{01} through R_{06} and the T-register. It should be used whenever linear regression or other built-in statistics functions are to be started. You can also use the routine at any time to clear these registers selectively without disturbing any other registers.

Solid State Software TI © 1978
BUSINESS DECISIONS MODULE CHECK BD-01
MODULE CHECK: SBR 2nd R/S
LINEAR REGRESSION INITIALIZATION: SBR CLR

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
	Library Module Check			
A1	Select Program		[2nd] [Pgm] 01	
A2	Run Module Check		[SBR] [2nd] [R/S]	9.[1]
	Initialize Linear Regression			
B1	Select Program		[2nd] [Pgm] 01	
B2	Initialize Linear Regression		[SBR] [CLR]	0.

NOTES: 1. The number 9. indicates the Business Decisions Library.

LONG-TERM FINANCING REQUIREMENTS

This program is designed as an aid in evaluating a company's long-term financing needs. The program allows you to project a company's financing requirements based on income, capital asset levels, the cost of capital for stocks and bonds, and the dividend rate. You can compare the future needs for bonds or stocks against your estimate of the demand and price for such issues in the future. With the help of this program, you can perform an analysis of a firm's growth and see the changes in financing needs based on management decisions.

To run the program you must have actual or estimated values for a number of variables for each period. You must enter the amount of retained earnings at the beginning of the period available for funding requirements in the period (E). The program assumes that there is cash or assets easily converted to cash available which is equal to E. You must also have the projected revenue for the period in dollars (REV), the projected percent gross profit of sales (%GPS), the total assets at the beginning of the period (TA), and the capital assets needed per sales dollar (CAS). The %GPS is determined by dividing gross profit by REV, where

Gross Profit = REV – Cost of Goods Sold – Selling and Administrative Expenses – Taxes

Note that GPS as defined here does not include taxes.

In addition you must specify the maximum percent of capital assets that is allowable for long-term debt (MD) and the percent of capital assets in long-term debt at the beginning of the first period (PD). Note that bonds are considered long-term debt, so PD is determined by dividing the value of outstanding bonds by the total assets at the beginning of the first period. You must also enter the projected dividend rate (DR), which is determined by dividing the projected GP into the amount of dividends in dollars that are projected to be paid during the period. Now you need just two more values: the cost of capital for bonds and the cost of capital for stocks, both expressed as a percent. The cost of capital for bonds (CB) can be determined with the help of program BD-03, *Debt Financing*, and the cost of capital for stocks (CS) can be determined with the help of program BD-04, *Investment Evaluation.*

The program follows a given precedence for financing the needed funding. First, the program uses retained earnings (E) to meet capital requirements before selling or issuing stocks and/or bonds. If retained earnings do not cover the necessary financing, then bonds and/or stocks are sold. Whether stocks or bonds are sold depends on the respective cost of capital for stocks and bonds. If bonds have a cost of capital lower than or equal to the cost of capital for stocks, then bonds are issued until the debt limit is reached. After the debt limit is reached, then stocks are sold.

The amounts shown for bonds and stocks sold or redeemed represent the cash amounts, not the par value. The program assumes that the necessary amount of bonds and stock can be sold at a price which holds the cost of capital for stocks and bonds constant. If the debt level at the beginning of the period exceeds the maximum allowable, then sufficient stock is sold to reduce the bonds outstanding to a suitable level based on the capital needs for the period.

When capital assets at the beginning of the period exceed the total required capital assets, funding during the period is not required. But the allowable long-term debt is computed on the current period's requirements and the existing long-term debt may exceed the new limit. As a result, stock is sold to reduce the debt to the current period's requirements, even though funding for assets is not necessary.

The amount of retained earnings at the end of a period can be reduced by increasing the dividend rate (DR) or by decreasing the ratio of capital assets to revenue (CAS).

Because of the dynamic nature of the stocks and bonds markets, projections must be updated frequently to reflect changing market conditions. With minimum inputs this program gives you a quick way to make estimates of financing requirements as market conditions change.

Solid State Software			TI © 1978	
LONG TERM FINANCING REQUIREMENTS				BD-02
% PD	% DR	% CB	% CS	E, Summary
REV	% GPS	CAS	% MD	TA

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 02	
2	Initialize		[SBR] [CLR]	0.00
3	Enter earnings applied to funding needed for period 1 of summary. **If none, enter 0.**[1]	E	[2nd] [E']	E†
4	Enter the following in any order:			
	a. Expected revenue in sales	REV	[A]	REV†
	b. % Gross profit of sales	%GPS	[B]	%GPS†
	c. Capital assets needed per sales dollar	CAS	[C]	CAS†
	d. % Maximum debt allowed	%MD	[D]	%MD†
	e. Present total assets[2]	TA	[E]	TA†
	f. % Present debt level[2]	%PD	[2nd] [A']	%PD†
	g. % Dividend rate	%DR	[2nd] [B']	%DR†
	h. % Cost of capital for bonds	%CB	[2nd] [C']	%CB†
	i. % Cost of capital for stock	%CS	[2nd] [D']	%CS†
5	Perform financial summary[3]		[2nd] [E']	period†*
	Gross Profit			P†*
	Dividends			D†*
	Retained Earnings			RE†*
	Capital Assets Needed			CAP†*
	Funding Needed			FN[5]†*
	Amount of Stock Sold			S†*
	Amount of Bonds Borrowed			B[6]†*
6	To continue the summary for subsequent periods, go to Step 4 and update inputs as necessary[4]			

NOTES:

1. If entry is miskeyed, go to Step 2.
2. No further inputs required after initial data is entered.
3. Be sure Step 4 inputs are correct before performing Step 5. Correct mistakes by reentering data in Step 4.
4. If an input error is discovered after execution of Step 5, start over at Step 2.
5. A negative value for FN indicates the amount by which the Capital Needed was exceeded by the Present Assets and Retained Earnings applied from the previous period. This amount is added to the retained earnings for the next period.
6. A negative value for B indicates the amount by which the present debt level exceeds the maximum debt allowed. This excess debt amount is liquidated by proportionately increasing the sale of stock, thereby ensuring that the present debt level equals the maximum debt allowed.

† These values are printed if the PC-100A is connected.

* These values are displayed for approximately 4 seconds.

TABLE OF CONTENTS

BD-02

Example: Determine the long-term financing requirements for the Sellmore Company during a four-year period beginning with 1979. The 1978 financial statement shows $1,290,000 in total assets with a 23% debt level. The 1978 earnings that can be applied to 1979 funding is $145,000. The sales goals for the four-year period are:

1979 –	$ 3,030,000
1980 –	5,330,000
1981 –	8,320,000
1982 –	10,820,000

The company expects a 10% gross profit from sales, a 25% upper limit on debt financing, and projects that $.84 per sales dollar will be needed in capital assets to meet its goals. The following information is available from company records.

Preferred Stock	
Par Value	$100 per share
Market Price	$70 per share
Contractual Dividend Rate	5.5%
Common Stock	
Market Price	$29 per share
Dividend	$.20 per share (annual)

The growth rate of common stock dividends is assumed to be 30% per year for 10 years. After 10 years, the growth rate is expected to stabilize at 6% per year. Examination of the capital stock base for 1978 showed that $40,000 was paid in common stock dividends and $10,000 was paid in preferred stock dividends. Net profit after taxes was $250,000; thus the dividend rate was (40,000 + 10,000) ÷ 250,000, or 20% of net income. It is desired to retain this 20% dividend rate over the four-year period.

The debt financing is handled by long-term bonds (30 years) having a coupon interest rate of 8% per year. The face value of the bonds is $1,000 and the market price is estimated at $900. The corporate tax rate is 48%.

Determine the type and amount of funding needed each year for the projected sales figures for the years 1979 – 1982.

Begin by examining respective costs of capital, based on the information given, for common and preferred stock. The lower cost of capital will be used. Then, compute the cost of capital for bonds.

1. Cost of Capital for Preferred Stock*:

$$k_p = \frac{Fd}{P} = \frac{100 \times 5.5}{70} \approx 7.9\%$$

**Quantitative Analysis of Financial Decisions,* James C.T. Mao, 1972, pp. 376 – 379.

where:

F = Face Value
d = Dividend Rate
P = Market Price

2. Cost of Capital for Common Stock:

   From example problem of Program BD-04, the cost of capital for common stock is 9.6%.

3. Cost of Capital for Bonds:

   From example problem of Program BD-03, the cost of capital for bonds is 4.6%.

These values and the other information given provide the inputs needed for Program BD-02.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT	
	[2nd] [Pgm] 02		Select Program		
	[SBR] [CLR]	0.00	Initialize		
145000	[2nd] [E′]	145000.00	Earnings	145000.00	
3030000	[A]	3030000.00	Sales	3030000.00	
10	[B]	10.00	GP	10.00	
.84	[C]	0.84	Cap/$ Sales	0.84	
25	[D]	25.00	% Max. Debt	25.00	
1290000	[E]	1290000.00	Total Assets	1290000.00	
23	[2nd] [A′]	23.00	% Pres. Debt	23.00	
20	[2nd] [B′]	20.00	% Div. Rate	20.00	
4.6	[2nd] [C′]	4.60	% Cb	4.60	
7.9	[2nd] [D′]	7.90	% Cs	7.90	
	[2nd] [E′]	"1."	Period	1.	
		"303000.00"	Gross Profit	303000.00	P
		"60600.00"	Dividends	60600.00	D
		"242400.00"	Retained Earnings	242400.00	RE
		"2545200.00"	Capital Needed	2545200.00	CAP
		"1110200.00"	Funds Needed	1110200.00	FN
		"770600.00"	Stock to Float	770600.00	S
		339600.00	Bonds to Sell	339600.00	B
5330000	[A]	5330000.00	Sales	5330000.00	
	[2nd] [E′]	"2."		2.	
		"533000.00"		533000.00	P
		"106600.00"		106600.00	D
		"426400.00"		426400.00	RE
		"4477200.00"		4477200.00	CAP
		"1689600.00"		1689600.00	FN
		"1206600.00"		1206600.00	S
		483000.00		483000.00	B

BD-02

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT	
8320000	[A]	8320000.00	Sales	8320000.00	
	[2nd] [E']	"3."		3.	
		"832000.00"		832000.00	P
		"166400.00"		166400.00	D
		"665600.00"		665600.00	RE
		"6988800.00"		6988800.00	CAP
		"2085200.00"		2085200.00	FN
		"1457300.00"		1457300.00	S
		627900.00		627900.00	B
10820000	[A]	10820000.00	Sales	10820000.00	
	[2nd] [E']	"4."		4.	
		"1082000.00"		1082000.00	P
		"216400.00"		216400.00	D
		"865600.00"		865600.00	RE
		"9088800.00"		9088800.00	CAP
		"1434400.00"		1434400.00	FN
		"909400.00"		909400.00	S
		52500.00		52500.00	B

Display values in quotation marks are flashed for a short time.

Register Contents

R_{00}	R_{05}	R_{10} REV	R_{15} DR	R_{20} RE	R_{25} Used
R_{01}	R_{06} E	R_{11} GPS	R_{16} PD	R_{21} D	R_{26} Used
R_{02}	R_{07} Used	R_{12} MD	R_{17} CAS	R_{22} P	R_{27} FN
R_{03}	R_{08} Used	R_{13} CB	R_{18} TA	R_{23} S	R_{28} Used
R_{04}	R_{09} Period	R_{14} CS	R_{19} CAP	R_{24} B	R_{29}

DEBT FINANCING

Using this program and Master Library Program ML-19, *Annuities*, you obtain the effective rate of interest after taxes on long-term debt such as bonds. Since the interest paid on long-term debt is tax deductible, long-term debt is often more favorable from a tax standpoint than other alternatives for financing needs.

A company pays less tax if it is paying interest on a bond. For example, suppose a company has paid $100 in interest during a year. If the company's corporate tax rate is 40%, then $40 less tax will be due at the end of the year. As a result of the interest deduction, the net cash paid for interest is $60. Therefore, long-term debt may be a desirable form of financing for a company because the tax reduction results in an effective interest rate that is lower than the coupon interest rate.

To use this program several values are needed. First you enter the number of periods until maturity (N) for the bond. Then you input both the market price of the bond (MP) and the face (par) value of the bond (FACE). Note that market price may be higher or lower than face value. In addition, you enter the coupon interest rate per period (Coupon). This is simply the interest rate of the bond divided by the number of payments per year. Also, you need the corporate tax rate (TAX) expressed as a decimal. This program then computes the after tax interest payment per coupon period (PMT factor) that is equivalent to the stated coupon interest payment. The coupon interest payment is equal to the coupon interest rate times the face value of the bond.

To obtain the effective interest rate, Program ML-19 in the Master Library is used. You must enter in Program ML-19 the same values for the number of periods (N), the market price (MP) and face value of the bond (FACE), as well as the PMT factor just computed. The effective interest rate computed by Program ML-19 is the value that is called the cost of capital for bonds used as input in Program BD-02, *Long-Term Financing Requirements.*

Solid State Software				TI ©1978
DEBT FINANCING				BD-03
				→ PMT factor
N	MP	FACE	Coupon	TAX

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 03	
2	Initialize		[SBR] [CLR]	0.
3	Enter Number of Periods	N	[A]	N
4	Enter Market Price of Bond (Net)	MP	[B]	MP
5	Enter Face Value of Bond	FACE	[C]	FACE
6	Enter Coupon Interest Rate per Period (decimal)	Coupon	[D]	Coupon (dec.)
7	Enter Corp. Tax Rate (decimal)	TAX	[E]	TAX (dec.)
8	Compute Payment Factor		[2nd] [E']	PMT factor
9	**Install Master Library Module**			
10	Select Program		[2nd] [Pgm] 19	
11	Initialize		[2nd] [E']	0.
12	Select Ordinary Annuity/PV		[2nd] [C']	0.
13	Enter N	N	[A]	N
14	Enter PMT Factor (Step 8)	PMT factor	[C]	PMT factor
15	Enter Market Price (Net)	MP	[D]	MP
16	Enter Face Value	FACE	[E]	FACE
17	Compute Cost of Capital	0	[B]	CB

NOTES: 1. Depressing the TRACE key on the PC-100A following Step 2 will provide a printout.

Example: Determine cost of capital for bonds for the Sellmore Company described in the example of Program BD-02. Face value of the bonds is $1,000 with market price estimated at $900. These 30-year bonds have annual coupons with an interest rate of 8% per year. Corporate tax rate is 48%.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 03		Select Program	
	[SBR] [CLR]	0.	Initialize	
30	[A]	30.	No. of Periods	
900	[B]	900.	Market Price	
1000	[C]	1000.	Face Value	
.08	[D]	0.08	Coupon %I (decimal)	
.48	[E]	0.48	Tax Rate (decimal)	
	[2nd] [E']	40.	Payment Factor	

INSTALL MASTER LIBRARY MODULE

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 19		Select Program	
	[2nd] [E']	0.	Initialize	
	[2nd] [C']	0.	Ord. Annuity/PV	
30	[A]	30.	No. Periods	
40	[C]	40.00	Payment Factor	
900	[D]	900.00	Market Price	
1000	[E]	1000.00	Face Value	
0	[B]	4.6228	Cost of Capital	

Register Contents

R_{00}	R_{05} TAX	R_{10}	R_{15}	R_{20}	R_{25}
R_{01} N	R_{06}	R_{11}	R_{16}	R_{21}	R_{26}
R_{02} MP	R_{07}	R_{12}	R_{17}	R_{22}	R_{27}
R_{03} FACE	R_{08}	R_{13}	R_{18}	R_{23}	R_{28}
R_{04} Coupon	R_{09}	R_{14}	R_{19}	R_{24}	R_{29}

BD-04

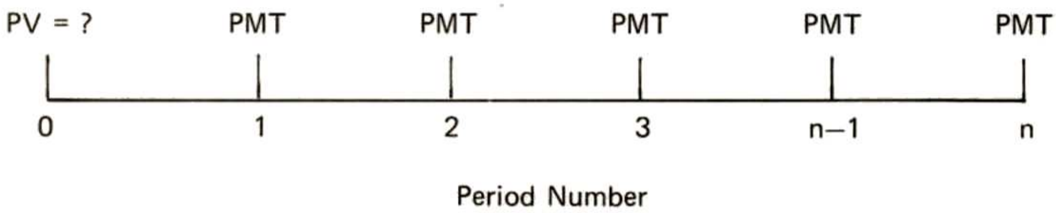
INVESTMENT EVALUATION

Many business decisions involve the time-value of money. Suppose a project is proposed with cash flows of varying amounts occurring over many years. You may want to find the return on the project and its net present value, given the firm's cost of capital rate. Suppose you need to decide whether to lease or purchase an asset based on the rate of return resulting from both alternatives. These are typical decisions requiring the use of a variable cash flow analysis. This program can aid you in making these decisions.

The program will compute the Present Value (PV), Future Value (FV), and Internal Rate of Return (IRR) for a series of variable cash flows. The IRR is the interest rate which makes the present value of the variable cash flows equal to the investment. If both positive and negative cash flows exist, there may be multiple IRR's. The program solves for a single IRR, which is generally the one closest to zero. The program will also compute the return on growth stock with different future dividend growth rates.

Present and Future Values of Variable Cash Flows

When present value is computed, the program assumes that the cash flows occur at the end of each payment period. The cash flow situation can be illustrated on a simple time line as shown here:

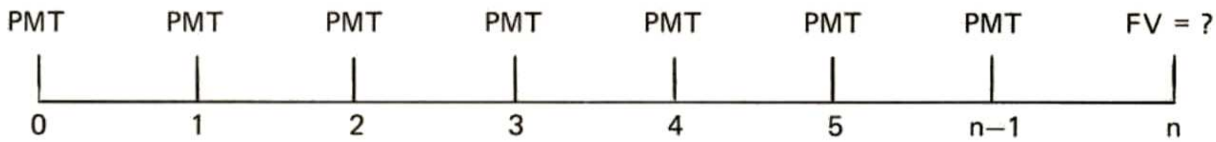


After the present value is calculated for payments made at the end of the period, you can calculate the present value for payments at the beginning of the period. The present value is converted by:

$$\text{PV (payments at beginning)} = (1 + i) \times \text{PV (payments at end)}$$

where i is the interest rate.

When computing future value, the program assumes payments occur at the beginning of the period.



You can calculate the future value for payments at the end of the period after you have calculated the future value for payments at the beginning of the period by:

$$\text{FV (payments at end)} = \frac{\text{FV (payments at beginning)}}{(1 + i)}$$

One problem which may also arise is a situation where the compounding period differs from the payment. For example, payments are made quarterly, and interest is compounded monthly. By converting the interest rate before entering it in the program, you can find the present value and future value with this program. The interest rate per payment period (γ) is given by:

$$\gamma = (1 + i)^{m/k} - 1$$

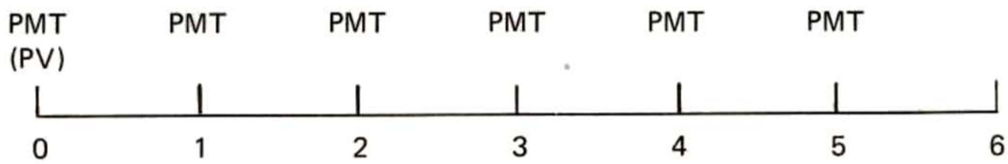
where i is the interest rate per compounding period, m is the number of compounding periods per year, and k is the number of payments per year. When γ is determined, enter it as the interest rate for present value and future value computations. If you need to shift the payments from the end of the period to the beginning of the period or from the beginning of the period to the end of the period, use the same equations described earlier, substituting γ for i.

Thus, with this program you can calculate present value and future value quickly and easily for a variety of situations. The program computes present value and future value when you enter a zero investment. If you enter a non-zero investment, the net present value is computed. Net present value is simply the present value minus the investment.

Internal Rate of Return

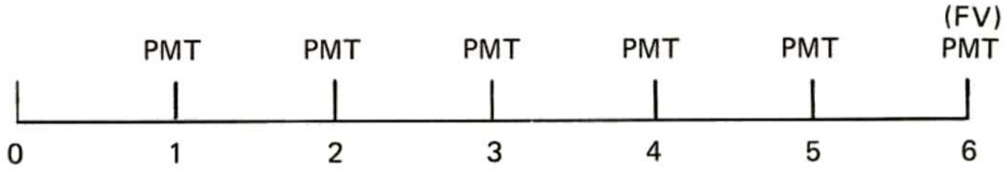
This program normally calculates the IRR for a given investment (PV) and a series of cash flows occurring at the end of each period. However, you can compute IRR for different situations with some modifications in the entry procedure. The following paragraphs outline the methods you can use when you have present value and cash flows at the beginning of the period or future value with cash flows either at the beginning or the end of the period.

For a given investment (PV) and payments at the beginning of the period, we have the following diagram.



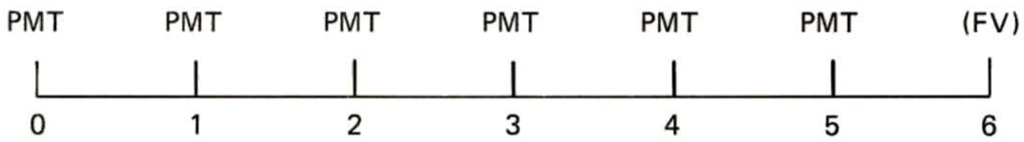
Notice that the first payment is made at the same time as the investment. To compute the IRR for this situation, add the first payment to the investment and enter the sum as the investment. Then the second payment is the first payment entered. After all the payments are entered, IRR is computed.

For a given future value with payments at the end of the period, we have the following diagram.



To compute IRR for this situation, enter a zero investment and enter all but the last payment normally. For the last payment, enter the negative of the sum of the last payment and the future value. Note that payment 6 occurs at the end of the last period, and therefore, is not compounded.

For a given future value and payments at the beginning of the period, the diagram becomes:



When payments are made at the beginning of the period, the given future value is for the end of the last period. To compute IRR, enter a zero investment. Then enter the payments normally. When all payments are entered, enter the negative of the future value as the last payment. Now the program will compute the correct IRR. Note that if you are using the PC-100A printer, the payment made at time zero is shown as payment one.

If you have a situation in which the payment period differs from the compounding period, you can obtain the correct IRR with the following equation.

$$i = (1 + \gamma)^{k/m} - 1$$

For this equation γ is the IRR per payment period, i is the equivalent IRR per compounding period, k is the number of payment periods per year, and m is the number of compounding periods per year.

Note that because of the nature of the internal rate of return calculation, there is some probability that more than one answer may exist to certain problems. In cases such as this, the program will in general find the rate of return closest to zero that satisfies the conditions you entered for the problem. In some cases, however, it is possible that no solution exists for a problem you enter. If after several minutes no solution has been found, press the [RST] key to stop the calculation.

Growth Stock Rate of Return

With specific information this program allows you to evaluate the return on a stock issue or purchase. To run the program you must know the price of the stock and the current dividend rate. You must also have an estimate of the supernormal dividend growth rate for n years and of the normal growth rate after n years. With this information you can calculate the return on a stock when it is expected to have a supernormal growth rate for a fixed number of years (n) and then drops to a normal growth rate. You may specify several different growth rates or individual dividend values during the time 1 to n.

Thus, this program allows you to analyze situations involving the present value, future value, and internal rate of return, where payments occur at the end or beginning of the period. You can also analyze situations where the compounding period differs from the payment period. This allows you to use this program in various situations involving variable cash flows such as: sinking funds, capital budgeting, leases, loans, mortgages, and wraparound mortgages. In addition, the program will aid you in analyzing the return on growth stock.

Solid State Software				TI © 1978
INVESTMENT EVALUATION				BD-04
CF# (Div#)	New CF (DIV)	Growth	Div ∞ → i	i → FV
INV (INC)	CF (DIV)	CF (DIV), N	→ i	i → PV

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
	Uneven Cash Flows			
1	Select Program		[2nd] [Pgm] 04	
2	Initialize		[SBR] [CLR]	0.
3	Enter Investment	INV	[A]	INV
4	Enter Cash Flows Using One or Both of the Following Methods:			
	I. Enter Cash Flow			
	• If Received	CF	[B]	CF
	• If Paid	CF	[+/–] [B]	–CF
	(Repeat as Needed)			
	II. a. Enter Expected Growth Rate of Cash Flows (decimal)	Growth	[2nd] [C']	Growth
	b. Enter 1st Cash Flow	CF	[C]	CF
	c. Enter Number of Cash Flows	N	[R/S]	CF_N
	(Repeat as Needed)			
5	To Change or Correct a Cash Flow Entry:			
	a. Enter Cash Flow No.	CF#	[2nd] [A']	CF#
	b. Enter Cash Flow			
	• If Received	New CF	[2nd] [B']	New CF
	• If Paid	New CF	[+/–] [2nd] [B']	–New CF
6	Compute Internal Rate of Return		[D]	i
7	Enter i As Decimal and Compute Present Value	i	[E]	PV
8	Enter i As Decimal and Compute Future Value	i	[2nd] [E']	FV
	Stock Flotation			
1	Select Program		[2nd] [Pgm] 04	
2	Initialize		[SBR] [CLR]	0.
3	Enter Current Value of Stock	INC	[A]	INC
4	Enter Dividend Payments Using One or Both of the Following Methods:			
	I. Enter Dividend Payment (Repeat as Needed)	DIV	[B]	DIV
	II. a. Enter Expected Growth Rate of Dividends As Decimal	Growth	[2nd] [C']	Growth
	b. Enter 1st Dividend Payment	DIV	[C]	DIV
	c. Enter Number of Payments (Repeat as Needed)	N	[R/S]	DIV_N

BD-04

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
5	To Change or Correct a Dividend Entry:			
	a. Enter Dividend No.	DIV #	[2nd] [A′]	DIV #
	b. Enter Dividend Payment	New DIV	[2nd] [B′]	New DIV
6	Enter Normal Growth Rate	Growth	[2nd] [C′]	Growth
7	Enter Normal Dividend and Compute Rate of Return	DIV_{∞}	[2nd] [D′]	i

NOTES: 1. 16+N data registers are required by the program. N is the number of cash flows or dividend payments.

Register Contents

R_{00} Used	R_{05} $\Sigma(-n)CF$	R_{10}	R_{15} Used	R_{20}	R_{25}
R_{01} Counter	R_{06} Used	R_{11} INV	R_{16} CF	R_{21}	R_{26}
R_{02} Used	R_{07} Used	R_{12} Used	R_{17} CF	R_{22}	R_{27}
R_{03} Growth	R_{08} Used	R_{13} Used	R_{18} CF	R_{23}	R_{28}
R_{04} ΣCF	R_{09} Pointer	R_{14} Used	R_{19} CF	R_{24}	R_{29}

NOTE: R_{16} and above are used for storage of cash flows or dividend payments.

Example 1: Your company has purchased a new machine. Its projected savings, net of taxes, for the next five years are:

Year	1	2	3	4	5
Savings	$1,000	$18,800	$21,000	$20,000	$19,600

You need to know the present value and the expected future value of the savings assuming that the savings occur at the end of each period. Your company expects a 16% annual return on all investments.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 04		Select Program	
	[SBR] [CLR]	0.	Initialize	
1000	[B]		1st Cash Flow	1.
		1000.00		1000.00
18800	[B]		2nd Cash Flow	2.
		18800.00		18800.00
21000	[B]		3rd Cash Flow	3.
		21000.00		21000.00
20000	[B]		4th Cash Flow	4.
		20000.00		20000.00
19600	[B]		5th Cash Flow	5.
		19600.00		19600.00
.16	[E]			0.16
		48664.98	PV (end of period)	48664.98
.16	[2nd] [E′]			0.16
			FV (beg. of period)	118567.18
	[X]	118567.18		
1.16	[1/X] [=]	102213.08	FV (end of period)	

Example 2: You must evaluate the proposed purchase of a new machine. It costs $40,500 and will generate the following after-tax savings each of the first five years.

Year	1	2	3	4	5
Savings	$450	$9,100	$11,500	$17,000	$24,500

What is the internal rate of return if the savings are realized at the end of each year.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 04		Select Program	
	[SBR] [CLR]	0.	Initialize	
40500	[A]	40500.00	Investment	40500.00
450	[B]		Ist Cash Flow	1.
		450.00		450.00
9100	[B]		2nd Cash Flow	2.
		9100.00		9100.00
11500	[B]		3rd Cash Flow	3.
		11500.00		11500.00

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
17000	[B]		4th Cash Flow	4.
		17000.00		17000.00
24500	[B]		5th Cash Flow	5.
		24500.00		24500.00
	[D]	0.120	IRR	0.120

Example 3: Determine cost of capital for common stock for the Sellmore Company described in the example of Program BD-02. Market price of the stock is $29 per share. Current annual dividend is $.20 per share, expected to grow at 30% per year for 10 years. After 10 years, the growth rate is expected to stabilize at 6% per year.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 04		Select Program	
	[SBR] [CLR]	0.	Initialize	
29	[A]	29.00	Market Price	29.00
.3	[2nd] [C']	0.30	Growth Rate	
.2	[C]	0.20	1st Dividend	
10	[R/S]		No. Periods	1.
				0.20
				2.
				0.26
				3.
				0.34
				4.
				0.44
				5.
				0.57
				6.
				0.74
				7.
				0.97
				8.
				1.25
				9.
				1.63
				10.
		2.12		2.12
.06	[2nd] [C']	0.06	Stable Growth Rate	
2.25*	[2nd] [D']		1st Normal Dividend	2.25
				0.060
		0.096		0.096

Cost of capital is 9.6%

*Stabilized growth rate was set at 6% beginning with the eleventh period. The first dividend during the normal growth period is:

($2.12 X .06) + $2.12 = $2.2472 or approx. $2.25

PROJECT PLANNING AND BUDGETING

This program allows you to store a matrix of data in the calculator. It is designed as an aid in building a 13-column spread sheet, useful in many planning operations. Up to 7(4) rows of data may be stored in the TI-59(58). The data may be entered either by entering each item individually or by specifying an amount for column 1 and a growth rate. Using the latter method, the calculator will compute and enter the values for columns 2 through 12 automatically. Once the data for columns 1 through 12 for a row has been entered, with a single command you can sum these elements and store the sum in column 13, or you can compute the average (sum/12) and store this value in column 13.

When all of the data has been entered you can manipulate the rows and columns. You can:

1. Add, subtract, multiply, or divide the column elements in two rows.
2. Add, subtract, multiply, or divide the column elements in a row by a constant.
3. Add and compute the average value of the elements in a column, and store the results in a new row.

The results of operations 1 and 2 can be stored in the calculator as a separate row, or if desired, can replace the data in one of the two existing rows. This versatility permits you to build a matrix which otherwise would be beyond the data storage capability of your calculator.

This program is especially useful for planning or budgeting situations where base values are stated with growth rates. The entry procedure requires only the base amount and growth factor. Thus, it is easy to perform sensitivity analysis or "what if" in such situations. As you become familiar with the program you will find many other uses for it.

Because of the inherently large amount of input and output data associated with this program, use of the printer is recommended.

Solid State Software			TI © 1978	
PROJECT PLANNING & BUDGETING				BD-05
Cnst, Row_B	MaxNo.Rows	Row, Col	Sh R, Data	Σ Col, $\bar{x}$Col
Row_A, Row_B	Row_C, RCL	Data, Growth	Sh L, Data	Σ Row, $\bar{x}$ Row

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 05	
2	Initialize		[SBR] [CLR]	0.00
3	Enter Maximum Number of Rows	Max No.	[2nd] [B′]	Max No.
4	Reset Partitioning	n[1]	[2nd] [Op] 17	Partitioning
	ENTER ELEMENTS 1–12 OF A ROW			
	Method A			
5	Enter Row Number	Row No.	[2nd] [C′]	Row No.
6	Enter Element i (Repeat for i = 1–12)	E_i	[C]	E_i
	Method B			
7	Enter Row Number	Row No.	[2nd] [C′]	Row No.
8a	Enter Element 1	E_1	[C]	E_1
8b	Enter Growth Rate[2] (decimal) and Complete Entry of Columns 2–12	Growth	[R/S]	Row No.[3]
	ENTER ELEMENT 13 OF A ROW (Following 1–12 Entry)			
	Method A			
9	Enter Element 13	E_{13}	[C]	E_{13}
	Method B			
10a	Compute and Store Sum of Elements 1–12 in Column 13	Row No.	[E]	ΣR
10b	Compute and Store Average of Elements 1–12 in Column 13		[R/S]	$\bar{x}$R
	ENTER SINGLE ELEMENT			
11a	Enter Row Number	Row No.	[2nd] [C′]	Row No.
11b	Enter Column Number	Col. No.	[R/S]	Col. No.
12	Enter Element[4]	E	[C]	E
	COLUMN TOTALS			
13a	Compute and Store Sum of Column Elements 1 Through (R–1) in Row R (See Step 3)	Col. No.	[2nd] [E′]	ΣC
13b	Compute and Store Average of Column Elements in Row R		[R/S]	$\bar{x}$C
	ROW OPERATIONS[3]			
14a	Enter Row A or Enter Constant	Row_A K	[A] [2nd] [A′]	Row_A K
14b	Enter Row B	Row_B	[R/S]	Row_B

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
15	Enter Resulting Row if Different From Row B[5]	Row_C	[B]	Row_C
16	Select Operation[6]			
	a. Add		[SBR] [+]	C_{13}[7]
	b. Subtract		[SBR] [–]	C_{13}[7]
	c. Multiply		[SBR] [×]	C_{13}[7]
	d. Divide		[SBR] [÷]	C_{13}[7]
	SHIFT OPERATIONS			
17a	Enter Row to be Shifted	Row_A	[A]	Row_A
17b	Enter Resulting Row[5]	Row_B	[R/S]	Row_B
18a	Enter Number of Locations Row is to be Shifted and			
	Shift Left or	No. Loc.	[D]	0.
	Shift Right	No. Loc.	[2nd] [D']	0.
18b	Enter New Data[8] (Repeat as Needed)	Data	[R/S]	Data
	RECALL A ROW			
19a	Enter Row Number	Row No.	[B]	Row No.
19b	Recall Row[3]		[R/S]	C_{13}

NOTES:

1. Set partitioning by pressing n [2nd] [Op] 17 according to the following:

No. of Rows	n
3	5*
4	6*
5	8**
6	9**
7	10**

*TI-58, within power-up partition for TI-59.
**TI-59 only.

2. Enter 0 for a constant value in each column.
3. Output format of the printer is

No.	Row	No.	Row
E_1		E_7	
E_2		E_8	
E_3		E_9	
$\sum_{i=1}^{3} E_i$	Σ	$\sum_{i=7}^{9} E_i$	Σ
E_4		E_{10}	
E_5		E_{11}	
E_6		E_{12}	
$\sum_{i=4}^{6} E_i$	Σ	$\sum_{i=10}^{12} E_i$	Σ

Observe that the data is organized by quarters for easy reference. Quarterly totals are also printed. If a printer is not available, the user should note that each of these values is displayed for approximately 2 seconds in the above order.

4. Note that at this point, Step 12 may be repeated to enter successive elements of a row. However, if too many entries are made the data will "spill over" into the next row.
5. The resulting row (row C) is where the new row is to be stored.

NOTES:

6. The sequence of operation is $R_A \square R_B = R_C$ or $K \square R_B = R_C$ (the $\square$ represents the selected operation). Note that if you want to subtract R_B from K, you select [SBR] [−]; however, if you want to subtract K from R_B you must enter −K and select [SBR] [+]. Similarly, to divide R_B by K, enter 1/K and select [SBR] [X].
7. In most cases C_{13} is simply the sum of elements 1–12. However, if the operation selected is $R_A \div R_B = R_C$, then C_{13} becomes the 13th element of R_A divided by the 13th element of R_B. If the operation is $K \div R_B = R_C$, C_{13} becomes the harmonic mean of the first 12 elements of the row ($12K \div \Sigma R$).
8. If the row is shifted left, the data is stored in the next available location. If it is shifted right, new data is stored from the front. A flashing display indicates an attempt to store data when the row is filled.

Register Contents

R_{00}	Used	R_{05}	Used	R_{10}		R_{15}		R_{20}		R_{25}	
R_{01}	Used	R_{06}	Used	R_{11}		R_{16}		R_{21}		R_{26}	
R_{02}	Used	R_{07}	Used	R_{12}		R_{17}		R_{22}		R_{27}	
R_{03}	Used	R_{08}	Data	R_{13}		R_{18}		R_{23}		R_{28}	
R_{04}	Used	R_{09}	Data	R_{14}		R_{19}		R_{24}		R_{29}	

NOTE: R_{08} through $R_{(7+13N)}$ are used for data storage.

Example: You are evaluating a project for your company and you need to prepare a project budget showing the gross profit under two different assumptions concerning unit sales. Unit variable cost is estimated to be $25 per unit, independent of the number of units produced. Total identifiable fixed cost is $25,000 per month. The projected sales performance is as follows for the two assumptions:

Sales Price	Unit Sales 1st Month	Unit Sales Growth
$40	2000	10%/month
$50	1800	5%/month

Compare the quarterly and annual sales, cost, and gross profit figures under these two conditions.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 05*		Select Program	
	[SBR] [CLR]	0.00	Initialize	
4	[2nd] [B']	4.00	Enter Max. Rows	
ASSUMPTION 1				
40	[X]	40.00		
2000	[=]	80000.00	Base Sales $	
1	[2nd] [C']	1.00	Row No.	
80000	[C]	80000.00	Base Sales	SEE TAPE 1
.1	[R/S]	1.00	Growth Rate	
1	[E]	1710742.70	Row 1 Sum	
25	[X]	25.00		
2000	[=]	50000.00	Base V. Cost	
2	[2nd] [C']	2.00	Row No.	
50000	[C]	50000.00	Base V. Cost	SEE TAPE 2
.1	[R/S]	2.00	Growth Rate	
2	[E]	1069214.00	Row 2 Sum	
3	[2nd] [C']	3.00	Row No.	
25000	[C]	25000.00	Base F. Cost	SEE TAPE 3
0	[R/S]	0.00	Growth Rate	
3	[E]	300000.00	Row 3 Sum	
1	[A]	1.00	Sales Row	
2	[R/S]	2.00	V. Cost Row	
4	[B]	4.00	Store in Row 4	
	[SBR] [–]	641528.51	(Sales – V.C.)	SEE TAPE 4
4	[A]	4.00	Sales – V.C.	
3	[R/S]	3.00	F.C.	
4	[B]	4.00	Store in Row 4	
	[SBR] [–]	341528.51	Gross Profit	SEE TAPE 5

*With the TI-58, press 6 [2nd] [Op] 17 to repartition before starting.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
ASSUMPTION 2				
50	[X]	50.00		
1800	[=]	90000.00	Base Sales $	
1	[2nd] [C']	1.00	Row No.	
90000	[C]	90000.00	Base Sales	SEE TAPE 6
.05	[R/S]	1.00	Growth Rate	
1	[E]	1432541.39	Row 1 Sum	
25	[X]	25.00		
1800	[=]	45000.00	Base V.C.	
2	[2nd] [C']	2.00	Row No.	
45000	[C]	45000.00	Base V.C.	SEE TAPE 7
.05	[R/S]	2.00	Growth Rate	
2	[E]	716270.69	Row 2 Sum	

Because base variable cost is one-half of base sales dollars, Row 2 is also sales minus variable cost.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
2	[A]	2.00	Sales – V.C.	
3	[R/S]	3.00	F.C.	
4	[B]	4.00	Store in Row 4	
	[SBR] [–]	416270.69	Gross Profit	SEE TAPE 8

TAPE 1		TAPE 2		TAPE 3		TAPE 4	
Sales		Var. Cost		Fixed Cost		Sales – F. Cost	
1.00	ROW	2.00	ROW	3.00	ROW	4.00	ROW
80000.00		50000.00		25000.00		30000.00	
88000.00		55000.00		25000.00		33000.00	
96800.00		60500.00		25000.00		36300.00	
264800.00	Σ	165500.00	Σ	75000.00	Σ	99300.00	Σ
106480.00		66550.00		25000.00		39930.00	
117128.00		73205.00		25000.00		43923.00	
128840.80		80525.50		25000.00		48315.30	
352448.80	Σ	220280.50	Σ	75000.00	Σ	132168.30	Σ
141724.88		88578.05		25000.00		53146.83	
155897.37		97435.86		25000.00		58461.51	
171487.10		107179.44		25000.00		64307.66	
469109.35	Σ	293193.35	Σ	75000.00	Σ	175916.01	Σ
188635.82		117897.38		25000.00		70738.43	
207499.40		129687.12		25000.00		77812.27	
228249.34		142655.84		25000.00		85593.50	
624384.55	Σ	390240.34	Σ	75000.00	Σ	234144.21	Σ
1710742.70	ΣR	1069214.19	ΣR	300000.00	ΣR	641528.51	

TAPE 5		TAPE 6		TAPE 7		TAPE 8	
Gross Profit		Sales		Var. Cost		Gross Profit	
4.00	ROW	1.00	ROW	2.00	ROW	4.00	ROW
5000.00		90000.00		45000.00		20000.00	
8000.00		94500.00		47250.00		22250.00	
11300.00		99225.00		49612.50		24612.50	
24300.00	Σ	283725.00	Σ	141862.50	Σ	66862.50	Σ
14930.00		104186.25		52093.13		27093.13	
18923.00		109395.56		54697.78		29697.78	
23315.30		114865.34		57432.67		32432.67	
57168.30	Σ	328447.15	Σ	164223.58	Σ	89223.58	Σ
28146.83		120608.61		60304.30		35304.30	
33461.51		126639.04		63319.52		38319.52	
39307.66		132970.99		66485.49		41485.49	
100916.01	Σ	380218.64	Σ	190109.32	Σ	115109.32	Σ
45738.43		139619.54		69809.77		44809.77	
52812.27		146600.52		73300.26		48300.26	
60593.50		153930.54		76965.27		51965.27	
159144.21	Σ	440150.60	Σ	220075.30	Σ	145075.30	Σ
341528.51		1432541.39	ΣR	716270.69	ΣR	416270.69	

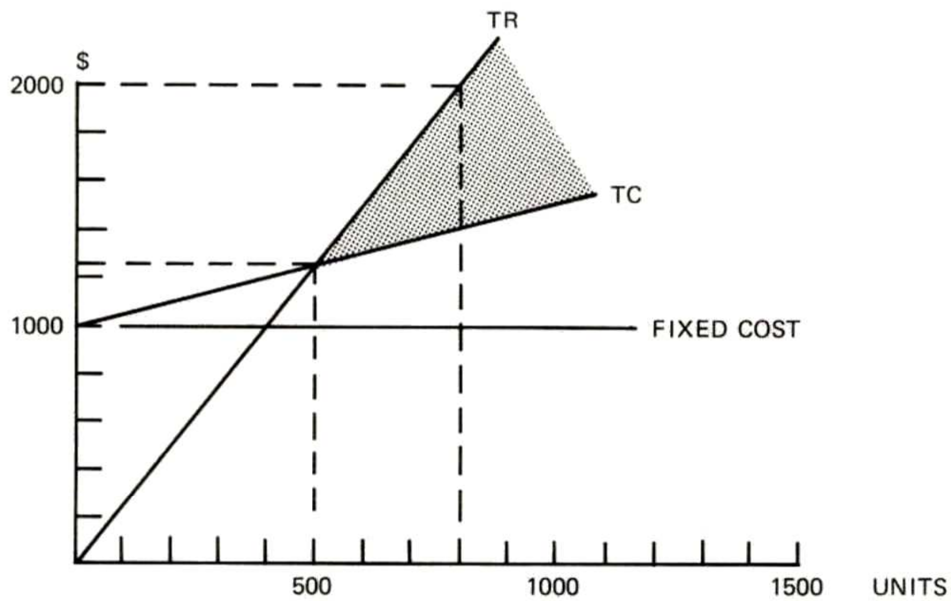
	1st Q	2nd Q	3rd Q	4th Q	Year
Assumption 1					
Sales	264800	352449	469109	624385	1710743
Variable Cost	165500	220281	293193	390240	1069214
Fixed Cost	75000	75000	75000	75000	300000
Gross Profit	24300	57168	100916	159144	341529
Assumption 2					
Sales	283725	328447	380219	440151	1432541
Variable Cost	141863	164224	190109	220075	716271
Fixed Cost	75000	75000	75000	75000	300000
Gross Profit	66863	89224	115109	145075	416271

BD-06

BREAKEVEN ANALYSIS

A useful tool for estimating the breakeven point and gross profit for a project is the Breakeven (Cost – Volume – Profit) Analysis. This program will allow you to do two different types of analyses. You can choose to assume either a constant or a learning curve rate for unit costs and selling prices. You will need to choose the analysis that best corresponds to your particular situation. To help you understand the two different analyses, we will work through some examples.

First we'll work an example that assumes a constant rate for unit costs and selling prices. Suppose a company is considering making a bid for producing a special fixture. The identifiable fixed costs (FIX CST) are $1,000. Identifiable fixed costs are those costs incurred that do not depend on the number of units produced and remain constant. In addition to the fixed costs, each unit has a variable cost (VAR CST) of $0.50. Variable costs are the incremental costs that are incurred for each unit that is produced. Each unit sells for $2.50. For this analysis we will assume that the unit price (U PRICE) of $2.50 and the unit variable cost of $0.50 remain constant regardless of the number of units produced. This assumption makes our analysis a linear model. If we graph the total revenue (TR) and the total cost (TC) we would see the graph shown below.



The shaded area represents gross profit. We can see that when TR = TC, the gross profit is zero. This is the breakeven point and in this example it is at 500 units. Thus if the company sells more than 500 fixtures, they will make a profit. From the graph we can see that the more units over 500 that are sold, the greater the gross profit. We can determine gross profit with the following equation:

Gross profit = Number of units sold X (U PRICE – VAR CST) – FIX CST

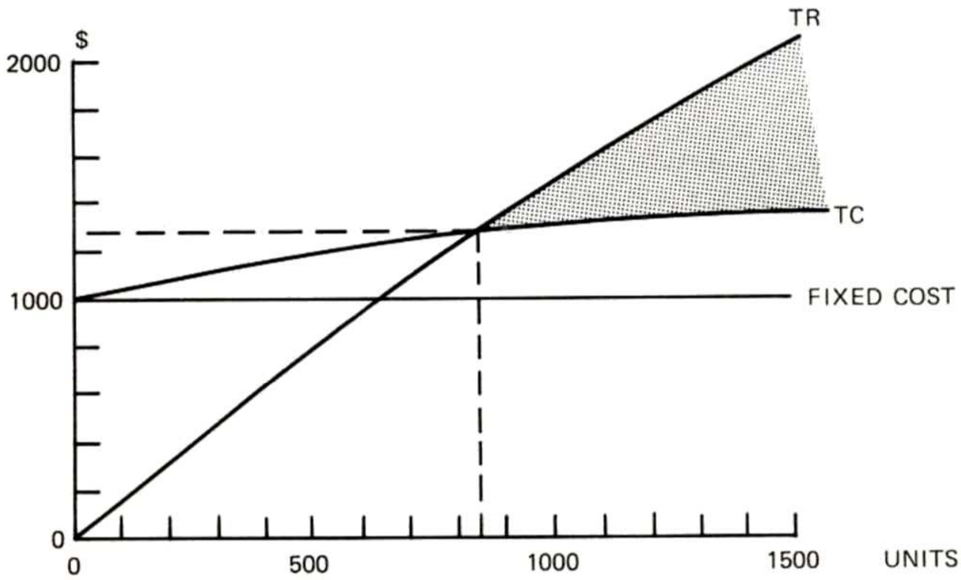
Thus if 800 units are produced and sold, then the gross profit is:

$$800\ (2.50 - 0.50) - 1000 = \$600$$

The assumption that the unit variable cost and unit price do not change with the number of units produced is quite often unrealistic. Since the unit variable cost and unit price often change as the production increases, you may want to assume *learning curve rates* for the variables. For this program a learning curve rate means that each time production and sales double then the cumulative average cost and revenue per unit changes by the amount of the learning curve.

Here is a short example to illustrate the idea. Assume that the initial sales price of $2.50 for the first fixture declines by 5% each time production doubles. Then if we sell 200 units the cumulative average sales price will have dropped to $1.68915 and our total revenue is $337.83 (1.68915 X 200 = 337.83). When we have sold 400 units, our cumulative average sales price is $1.6046925/unit (1.68915 X .95 = 1.6046925). At this sales price our total revenue is $641.88 (1.6046925 X 400 = 641.88). Note that it is the average price for all units sold that is reduced by the learning curve. The same method may be applied to variable cost.

If a learning curve is used in the breakeven analysis then total revenue and total cost appear as curved lines in the graph. The breakeven point still occurs where TR = TC and the gross profit is zero. We can see this in the sample graph shown below.



The shaded area represents gross profit.

BD-06

This program allows you to enter a learning rate for variable costs that is either positive or negative. A positive learning rate means unit costs will increase as production increases. A negative learning rate means costs will decrease as production increases. You may also enter a positive or negative learning rate called erosion rate here for unit sales prices. A positive rate means sales prices increase as sales increase and a negative rate means sales prices decrease as sales increase. When the learning rate is zero, a constant value is assumed for unit cost. A constant value for the unit sales price is assumed when the erosion rate is zero.

Several assumptions are made in the breakeven analysis that may cause a difference between the actual results and the results projected by the analysis. It is assumed in the analysis that all units produced are sold and that there are no changes in unit inventory levels. We can see from the graphs that the breakeven point in dollars is always above the fixed dollar costs. The fixed costs included are those costs incurred because of the project and should not include any allocated fixed costs that existed before the project started. We must assume that the unit price and variable cost estimates are valid within a relevant range. Note that the program assumes all fixed costs are recovered before a profit can be made.

With this program you can perform a cost — volume — profit analysis using either of the two approaches discussed. Because of the assumptions and estimated unit prices and variable costs, you should perform a sensitivity analysis on a project. Using different cost and price estimates you can observe the relative effect of the changes on gross profit at projected sales if variable costs and unit prices are either underestimated or overestimated. If actual costs and prices are very different from those projected, the actual breakeven point could be much higher or lower than expected.

To run the program you need to know the fixed costs, the estimated variable cost per unit, and the estimated unit price. Given these estimates, the total cost and total revenue will be computed for a specified number of units produced and sold. The necessary number of units to be produced and sold to earn a specified percent Gross Profit Margin (GPM) can also be computed. Note that a 0% GPM will yield the breakeven point in units. Conversely, given the number of units produced and sold, the program will compute the GPM. Also, for nonlinear costs and revenues, the program will compute the maximum GPM if it exists. We can see from the graphs that if the costs and revenues are linear, there is no maximum gross profit.

This program computes many important values necessary in doing a breakeven analysis. By trying different costs and prices you can see the effects of inaccurate estimates. This will give you a picture of how much your gross profit could vary. With this information you can then make the necessary decisions to achieve the desired results.

Solid State Software			TI ©1978	
BREAKEVEN ANALYSIS				BD-06
		START	COMMAND	UNIT/GPM
VAR CST	LEARN	FIX CST	U PRICE	EROSION

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 06	
2	Initialize		[SBR] [CLR]	0.
3	Input In Any Order:			
	a. Variable Cost ($/unit)	VAR CST	[A]	VAR CST
	b. Cost Learning Rate (%)[1]	LEARN	[B]	100 + LEARN
	c. Fixed Cost ($)	FIX CST	[C]	FIX CST
	d. Unit Price ($/unit)	U PRICE	[D]	U PRICE
	e. Price Erosion Rate (%)[1]	EROSION	[E]	100 + EROSION
	f. Command as Follows:	COMMAND	[2nd] [D']	COMMAND
	1—Find number of units to generate specified GPM. 2—Find GPM from sale of specified number of units. 3—Find total cost to produce specified number of units. 4—Find total revenue from selling specified number of units. 5—Find units for maximum GPM			
	g. Enter GPM% if Command is 1 Enter units if command is 2, 3, or 4. Not required if command is 5.	UNIT/GPM	[2nd] [E']	UNIT/GPM
4	Start Calculation		[2nd] [C']	RESULT

NOTES:

1. Constant percentage change in unit cost (price) for learning curve or erosion effect.
2. All dollar amounts are displayed to the nearest cent.
3. All unit outputs are displayed to the nearest unit.
4. All percentage outputs are displayed to the nearest hundredth of a percentage point.
5. Error indications (flashing display):
   - a. No maximum GPM.
   - b. GPM specified greater than maximum.
   - c. GPM specified greater than or equal to 100%.
   - d. Learning or Erosion Rate $\leqslant -50\%$.
6. Taxes are ignored in this program.

BD-06

Example: A small manufacturer is planning to produce an item which he expects to introduce at $2.50 each. This item will cost $.50 each to produce. In addition, it will cost $1,000 to set up for production. He expects his cost per unit to *increase* on a 5% learning curve. Further, he expects the selling price to *decrease* on a 5% erosion curve. How many units must he produce to break even? What is his total revenue (and cost) at the breakeven point? How many units must he produce to maximize his gross profit margin (GPM)? What is his maximum GPM? Find his total cost and total revenue at this GPM level. How many units will yield a GPM of 10%?

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 06		Select Program	
	[SBR] [CLR]	0.	Initialize	
.5	[A]	0.5	Variable Cost	
5	[B]	105.	Learning Rate	
1000	[C]	1000.	Fixed Cost	
2.5	[D]	2.5	Unit Price	
5	[+/–] [E]	95.	Erosion Rate	
1	[2nd] [D′]	1.	Units at GPM	
0	[2nd] [E′]	0.	GPM	
	[2nd] [C′]	1663.	Units to Breakeven	
1663*	[2nd] [E′]	1663.	Units at Breakeven	
4	[2nd] [D′]	4.	Revenue at Units	
	[2nd] [C′]	2401.34	Rev. at Breakeven	
5	[2nd] [D′]	5.00	Units for Max. GPM	
	[2nd] [C′]	6886.		
6886*	[2nd] [E′]	6886.	Units	
2	[2nd] [D′]	2.	Find GPM	
	[2nd] [C′]	17.18	Max. GPM	
3	[2nd] [D′]	3.00	Find Cost	
	[2nd] [C′]	7413.19	Cost	
4	[2nd] [D′]	4.00	Find Revenue	
	[2nd] [C′]	8951.22	Revenue	
10	[2nd] [E′]	10.00	Desired GPM	
1	[2nd] [D′]	1.00	Find Units	
	[2nd] [C′]	2565.	Units	

*This value is already in the display at this point.

Register Contents

R_{00}	R_{05} Used	R_{10} VAR CST	R_{15} COMMAND	R_{20} Used	R_{25}
R_{01}	R_{06} Used	R_{11} LEARN	R_{16} UNIT/GPM	R_{21}	R_{26}
R_{02}	R_{07} Used	R_{12} FIX CST	R_{17} f	R_{22}	R_{27}
R_{03}	R_{08} Used	R_{13} U PRICE	R_{18} g	R_{23}	R_{28}
R_{04} Used	R_{09} Used	R_{14} EROSION	R_{19} Used	R_{24}	R_{29}

FACILITY CAPACITY

In many organizations there are processes which generate waiting lines or queues. Waiting lines occur when someone must wait for service because the servicing facility is temporarily unable to provide that service. A simple example of a queue is the checkout counters in a supermarket. The manager of the supermarket may add more checkers in order to reduce the time the customers must wait for service. However, he knows that he cannot afford to have enough checkers to guarantee no waiting line. So the manager tries to balance the situation so that he has enough checkers to keep the customers happy, but not more checkers than he can afford. This situation is also found often in industry but with another twist. Industry typically pays for both the lost time of persons waiting in line as well as the wages of the people who provide the service. Queuing theory provides a way for management to arrive at the lowest total cost by determining how many persons to hire for the servicing facility.

Three basic situations may be analyzed with this program. The first situation is a single chain of servers in series, with an exponential service time distribution. In this situation customers would need to pass through several different service centers. An example of this situation would be a cafeteria line. The other two situations both use multiple servers in parallel. One of these assumes an exponential service time distribution and the other a constant service time distribution.

In order to work the queuing problem, we make some basic assumptions. First, we assume that customers are continually arriving and that the arrival times of customers are Poisson distributed. We also assume that customers are served on a first come, first served basis and that no customer leaves the queue without being served. In many cases customer arrivals are Poisson distributed and service times are exponentially distributed. A common exception occurs when machines with automated cycles (such as vending machines) are involved – where service times are constant.

In the Poisson distribution, the probability of an arrival is independent of the arrival of any other customers, and the arrivals are considered to be at random. The Poisson and exponential distributions are related in a special way. If the *number of arrivals* has a Poisson distribution, then the *times between arrivals* are exponentially distributed. Because of this unique relationship, if the *times required to serve customers* are exponentially distributed, then the *number of completions of service* has a Poisson distribution. Thus in the multiple server situation with an exponential service time distribution, the customer arrival and departure times are Poisson distributed.

In order to use this program the user must define the average number of customer arrivals per time period (λ), the average number of customers served per time period (μ), and the number of servers (C) if the parallel system is assumed, or the number of phases every customer must pass through (K) if the series system is assumed. Given these inputs the program provides the expected number of customers in the system (L), and the expected number of customers in the queue (Lq) at any given time as well as the expected waiting time per customer (Wq) and the expected total time in the system per customer (W). Also computed is the utilization factor of the servers (UTIL). The utilization factor is the fraction of time the servers are being used.

If certain cost data is provided, this program also determines the total cost per customer for waiting time (TCW) and the total cost for facility idle time (TI). The idle time cost per customer may be computed as TCI = TI/L. The inputs required to obtain these results are the cost per time period of customer waiting time (CW), and of the facility idle time (CI).

You can use this program to help you minimize total cost by trying different numbers of servers. The only restriction is that for the parallel server case, the product of the number of servers (C) and the customers served per time period (μ) must be greater than the customer arrivals per time period (λ). For the series case, the customers served per time period (μ) must be greater than the customer arrivals per time period (λ). This is necessary to ensure that the servers can keep up with the arrival of customers. There are many cases where you may want to minimize or maximize some of the other variables. This can also be done using different numbers of servers. In general, by varying the number of servers you can minimize the expected waiting time per customer thereby reducing the expected total time in the system for each customer. You can also experiment to see the effect of increasing the number of customers to the facility or decreasing the average service time per customer. If you are not sure of the service time distribution for parallel servers you may compute results for both exponential and constant service times. The exponential case provides a pessimistic estimate of performance while the constant case is optimistic. Thus, the program can provide you with helpful information for optimizing the capacity of a facility.

References:

Gross & Harris, *Fundamentals of Queuing Theory*, John Wiley 1975.

Molina, E.C., "The Theory of Probabilities Applied to Telephone Trunking Problems." *Bell System Technical Journal*, Volume 1, 1922, pp 69-81.

Molina, E.C., "Application of the Theory of Probability to Telephone Trunking Problems." *Bell System Technical Journal*, Volume 6, 1927, pp 461-494.

Saaty, T.L., *Elements of Queuing Theory*, McGraw-Hill, 1961.

Syski, A., *Introduction to Congestion Theory in Telephone Systems,* Oliver & Boyd, 1960.

Solid State Software				TI ©1978
FACILITY CAPACITY				BD-07
OPT	CI			Compute (D)
λ	μ	C or K	CW	Compute (P)

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 07	
2	Initialize		[SBR] [CLR]	0.
3	Input In Any Order:			
	a. Arrivals per time period	λ[1,2]	[A]	λ
	b. Service rate	μ	[B]	μ
	c. Number of servers, or	C[3]	[C]	C
	Number of phases	K[4]	[C]	K
	d. Cost per time period of customer waiting time	CW	[D]	CW
	e. Cost per time period of facility idle time	CI	[2nd] [B']	CI
	f. Option desired:	OPT	[2nd] [A']	OPT
	−1—Series case 0—Parallel case (expon) 1—Parallel case (const)			
4	Compute with Printer, or		[E]	
	Compute without Printer		[2nd] [E'][8]	L Lq W Wq TCW[5] TI[5,6] See Note 7[5] UTIL

NOTES:

1. $\lambda < C\mu$ for parallel server case.
2. $\lambda < \mu$ for series server case.
3. $C > 0$
4. $K > 0$
5. This output can be ignored if cost data was not entered.
6. Divide this result by the number of customers in the system (L) to determine idle cost per customer (TCI).
7. This item is misstated due to the idle cost component. To correct it, TCI, obtained per Note 6 should be added to the waiting cost (TCW) to compute the total cost (TC).
8. Press [R/S] to obtain subsequent results without printer.

Example 1: New employees of the Apex Company must do several things on their first day before they actually begin work. First they must fill out payroll and insurance forms, and have an identification badge made. After that they are fitted for safety equipment and are issued a tool kit. Thus, there are four phases each employee must pass through. On the average seven new employees arrive each hour. Each phase can serve 10 employees per hour. The cost of facility idle time is $24 per hour, and the new employees' time costs the company an average of $8 per hour. Determine the average time required to process each new employee. How much does it cost the company to process each new employee?

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 07		Select Program	
	[SBR] [CLR]	0.	Initialize	
7	[A]	7.	Enter λ	
10	[B]	10.	Enter μ	
4	[C]	4.	Enter k	
8	[D]	8.	Enter CW	
24	[2nd] [B′]	24.	Enter CI	
1	[+/−] [2nd] [A′]	−1.	Select Series Case	
	[E]		Customers in system	9.3333
			Customers in queue	6.5333
			Time in system	1.3333
			Time in queue	0.9333
			Waiting cost	7.4667
			Idle factor	9.6000
			Ignore	17.0667
		0.7000	Utilization	0.7000

The processing time is 1.3333 hours per employee. The facility idle cost per employee is obtained by dividing the idle factor by the customers in the system (9.6 ÷ 9.3333 = 1.0286). This number added to the waiting cost per employee yields the total cost per employee (1.0286 + 7.4667 = 8.4953).

Example 2: An automotive repair shop has a parts department where the mechanics obtain parts. There are three men working at the parts counter and they each serve eight mechanics per hour. Suppose mechanics come to request parts on the average of 18 mechanics per hour. The cost of a parts man being idle is $5 per hour and a mechanic being idle costs $9 per hour. Assume exponential service time with multiple servers in parallel. What is the cost per mechanic served under these conditions? What is this cost if another parts man is added?

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 07		Select Program	
	[SBR] [CLR]	0.	Initialize	
18	[A]	18.	Enter λ	
8	[B]	8.	Enter μ	
3	[C]	3.	Enter c	
9	[D]	9.	Enter CW	
15	[2nd] [B′]	15.	Enter CI	
0	[2nd] [A′]	0.	Select par case, exp. serv.	

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[E]		L	3.9533
			Lq	1.7033
			W	0.2196
			Wq	0.0946
			TCW	0.8516
			TI	0.8236
			Ignore	1.6752
		0.7500	UTIL	0.7500
4	[C]	4.0000	Enter c	
20	[2nd] [B']	20.0000	Enter CI	
	[E]		L	2.5601
			Lq	0.3101
			W	0.1422
			Wq	0.0172
			TCW	0.1550
			TI	1.2445
			Ignore	1.3995
		0.5625	UTIL	0.5625

With three parts men the cost is $1.0599 per mechanic (0.8236 ÷ 3.9533 + 0.8516 = 1.0599). The cost per mechanic is $0.6411 with four parts men (1.2445 ÷ 2.5601 + 0.1550 = 0.6411).

Example 3: A company has an automated inventory system where employees can personally select the parts they need. However, they must also key the pertinent information into a computer which tallies the inventory entries. Each time an employee uses a computer terminal, his time on the terminal is very close to 5 minutes. Thus, we can assume a *constant service rate* of 12 per hour for each terminal. Currently three terminals are in use which cost the company $1,000 each per month, or about $5.70 per hour [$1,000/(22 × 8) ≈ $5.70]. The average salary of the employees that use the terminals is $4.50 per hour. An average of 30 employees per hour come to check out parts. How much will adding another terminal lower the cost per employee served?

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 07		Select Program	
	[SBR] [CLR]	0.	Initialize	
30	[A]	30.	Enter λ	
12	[B]	12.	Enter μ	
3	[C]	3.	Enter c	
4.5	[D]	4.5	Enter CW	
3	[×]	3.		
5.7	[=] [2nd] [B']	17.1	Enter CI	
1	[2nd] [A']	1.	Parallel-Const.	
	[E]		L	5.7363
			Lq	3.2363
			W	0.1912
			Wq	0.1079
			TCW	0.4854
			TI	0.5450
			Ignore	1.0304
		0.8333	UTIL	0.8333

BD-07

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
4	[C]	4.0000	Enter c	
4	[X]	4.0000		
5.7	[=] [2nd] [B']	22.8000	Enter CI	
	[E]		L	2.9553
			Lq	0.4553
			W	0.0985
			Wq	0.0152
			TCW	0.0683
			TI	0.8423
			Ignore	0.9105
		0.6250	UTIL	0.6250

With three terminals the total cost is 0.5450 ÷ 5.7363 + 0.4854 = $0.5804. With four terminals total cost is 0.8423 ÷ 2.9553 + 0.0683 = $0.3533. The cost saving per employee served is $0.5804 – $0.3533 = $0.2271.

Register Contents

R_{00}	R_{05} λ	R_{10} Used	R_{15} OPT	R_{20} Used	R_{25}
R_{01} Used	R_{06} μ	R_{11} Used	R_{16} Wq	R_{21} TCW	R_{26}
R_{02} Used	R_{07} C or K	R_{12} CW	R_{17} W	R_{22} TI	R_{27}
R_{03} Used	R_{08} ρ	R_{13} Used	R_{18} Lq	R_{23} TC	R_{28}
R_{04} Used	R_{09} ρc	R_{14} CI	R_{19} L	R_{24}	R_{29}

ECONOMIC REORDERING AND PRODUCTION RUNS

All businesses which produce or purchase merchandise have inventory costs. Since companies cannot sell merchandise as soon as it is produced or purchased, they must carry an inventory and hence incur costs associated with it. Inventory difficulties can and do contribute to business failures. If a retail merchant does not have merchandise in stock when a customer requests it, methods must be employed to keep the customer happy with his service. If the customer chooses to purchase the merchandise from a competitor, then you have incurred a cost from this lost sale. However, carrying sufficient inventory to meet all possible requests can result in excessive carrying costs that could mean the difference between making a profit and showing a loss. Thus, although inventory costs are necessary, they must (and can) be minimized for the most profitable business conditions.

This program is designed to aid you in determining the most economic inventory level and when to start a new production run or place a new order. This program determines that order quantity (EOQ) and reorder point (R) or that production quantity (EPQ) and reorder point (R) which minimizes total annual cost (TAC). You may also define discontinuities (price breaks) in the TAC versus order quantity or production quantity function. Minimum TAC may be found for either EOQ or EPQ under either the constraint of a stated unit cost of a stockout (St), or under the constraint of a stated probability of stocking out (Pr). The input and output may be interpreted in the same fashion whether you are finding EOQ or EPQ.

Because the program uses cost inputs which are not usually carried in a company's accounting system, you must often estimate such costs. Fortunately, errors in estimating the costs are usually not serious because the minimum TAC is relatively insensitive to slight variations in the cost estimates. For your input data you can determine the sensitivity by varying the inputs. If varying the inputs shows small changes in the output, then the model is insensitive for your values. However, if there are large variations in the output, the model is sensitive to the values you are using. If the model is sensitive, you will want to estimate your input values as carefully as possible.

To determine EOQ or EPQ you must have values for many variable costs. You must know or estimate the number of items needed or demand per year (DPY) and the cost per unit (CPU). Many times the cost per unit will decrease as the number of units ordered increases. This change in cost per unit is called a *price break*. This program can handle price breaks and you can enter each price and the minimum order quantity required to receive that price. Note that if there is only one price regardless of how many units are ordered, then there would be one price break with no minimum units required to get that price. If you are using the program for a production model, you must specify a production quantity per year (P) which must be greater than DPY.

In addition you need the *cost for processing each order* (CPO). Ordering costs include any costs involved in getting an item into inventory. These may include: the cost of reviewing inventory decisions, clerical costs associated with placing an order and handling the accounting, and the labor costs to put the merchandise into stock when it is received. You also need the annual *unit holding cost* (UHC) as a decimal percent of the cost per unit. Holding costs can include such things as: cost for renting space and maintaining storage facilities, record keeping of inventory, insurance on the inventory, taxes, costs for obsolete or spoiled inventory, and the cost equal to what the money could be earning if it were invested in something other than inventory. Be sure to include all costs incurred because of carrying inventory, even if the costs do not appear on any financial statements. Note that all costs should be in the same units, e.g., dollars.

You must also have an estimate of the expected demand during lead time (EDDLT) in units, and the standard deviation of the demand during lead time (σ). *Lead time* is the time interval between ordering goods and receiving those goods. In general, the demand during lead time will not always be constant. Thus, if you have several sample values for demand during lead time, simply average those values to obtain the expected demand during lead time. Then you can use those same values to obtain the standard deviation of the demand during lead time.

To clarify all this, let's consider an example. Suppose you have the following demand data for the past year:

Month	Demand (units)	
January	150	
February	170	
March	140	
April	130	
May	180	
June	150	
July	140	
August	170	
September	130	
October	140	$\overline{x}$ = 149.2
November	130	σ = 16.6
December	160	

If we assume that the lead time is one month, then we can choose 150 units for EDDLT with a standard deviation of 16.6. Remember that the standard deviation of the data does not change, even if you choose to use a larger EDDLT. Note that we assume demand level is constant, i.e., there is no trend or seasonality in the data. If you have highly seasonal data, you may wish to do the analysis for each season.

One additional input is needed. If you know the stockout cost per unit (St), this is the only other value you need. The *stockout cost* per unit is the actual amount it costs the firm if they are out of stock on an item. Stockout cost may include additional paperwork necessary to backorder an item, time lost by the person or department requiring the item, and possible costs incurred if the customer obtains the item from a competitor.

Many times, however, it is difficult to determine the cost per unit for being out of stock. For example, in a manufacturing company if being out of stock produces a bottleneck on the assembly line, the value of this is hard to assess. Also, the stockout cost per unit is often not constant. In these situations, you may wish to enter desired probability of stocking out during lead time (Pr). This is the maximum percentage of time you want to be out of stock. For example, if you want to be in stock 95% of the time then Pr would be 0.05 (1.0 – 0.95 = 0.05). The program will allow you to enter either Pr or St.

Given the appropriate input values, this program will solve for R and the EOQ or EPQ that minimizes total annual cost. The *reorder point* (R) is given in units and is the number of units left in inventory when you should reorder. The *EOQ or EPQ* then is the number of items you should order or begin producing at that point. The program will also compute either Pr or St, whichever you did not input. In addition, the program will compute the expected number of units greater than R required during the lead time [E(DDLT > R)]. The user may determine the safety stock (SS) as R–EDDLT. Safety stock is the extra inventory held against the possibility of a stockout.

Inventories add operating flexibility to organizations that they would not otherwise have. Since inventory is often the largest current asset of the organization, inventory control deserves the attention of management. Good inventory management can make a significant contribution to a firm's profit showing and this program can help in that management responsibility.

Reference:

McMillan, Claude and Gonzalez, Richard L., *Systems Analysis: A Computer Approach to a Decision Model*, Richard D. Irwin, Inc., 1968, pp 175-189, 198-202.

Solid State Software			TI ©1978	
ECONOMIC REORDERING & PROD. RUNS				BD-08
DPY	EDDLT	σ	P	Compute
CTL;Qy,St,Pr	N	Q_i ; CPU_i	UHC	CPO

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 08	
2	Initialize		[SBR] [CLR]	0.
3	Repartition if Necessary[1]	No. Reg./10	[2nd] [Op] 17	
4	Enter Number of Price Breaks[2]	n	[B]	0.
5a	Enter Quantity at i^{th} Price Break[3]	Q_i	[C]	0.
5b	Enter Unit Price at i^{th} Price Break	CPU_i	[R/S]	0.
5c	Repeat 5a and 5b for n Pairs (Q_i, CPU_i)			
6	Enter Unit Holding Cost[4]	UHC	[D]	UHC
7	Enter Cost/Order Placed	CPO	[E]	CPO
8	Enter Demand Quantity/Year	DPY	[2nd] [A']	DPY
9	Enter Expected Demand During Lead Time	EDDLT	[2nd] [B']	EDDLT
10	Enter Std. Dev. of EDDLT	σ	[2nd] [C']	σ
11	Specify Type of Run[5]	CTL	[A]	0.
12	Enter Data	See Note 5	[R/S]	Data
13	Enter Prod. Quantity/Year[6]	P	[2nd] [D']	P
14	Enter Stockout Cost[7]	St	[STO] 17	St
15	Enter Probability of Stockout[7]	Pr	[STO] 15	Pr
16	Start Computation		[2nd] [E']	TAC[8]
			[R/S]	EOQ or EPQ
			[R/S]	R
			[R/S]	E(DDLT > R)
			[R/S]	Pr
			[R/S]	St

NOTES:

1. Each price break requires two data registers beginning with R_{29}. The TI-59 will handle up to 15 price breaks with power-up partitioning (479.59). The TI-58 must be repartitioned as follows:

	No. of Price Breaks:	Press:
follows:	1–5	4 [2nd] [Op] 17
	6–10	5 [2nd] [Op] 17
	11–15	6 [2nd] [Op] 17

2. n must be ≥ 1 and must be immediately followed by Q_1 on input.
3. Q_1 must be 0.
4. UHC is a decimal fraction of CPU
5. **CTL**

CTL	Type of Run	Data Entered After CTL
0	Calc. TAC	Order or Production Qty.
1	Calc. EOQ w/known St	St
2	Calc. EOQ w/known Pr	Pr
−1	Calc. EPQ w/known St	St
−2	Calc. EPQ w/known Pr	Pr

6. P is always required if CTL is −1 or −2. Required for CTL = 0 if TAC is to be calculated for a given production quantity.

NOTES: 7. Steps 14 and 15 are required only if CTL = 0. Pr or St must be 0, depending on whether TAC is to be calculated with a known cost of stockout or with a known probability of a stockout.

8. Output values are printed if the PC-100A is connected.

Error Conditions

1. $n \neq$ number of (Q_i, CPU_i) pairs
2. Illegal CTL digit
3. $Q_1 \neq 0$
4. For CTL = 0, both Pr and St = 0
5. Calculated Pr $\geqslant$ 1
6. St = 0 for CTL = ±1
7. Pr outside the range 0 to 1 for CTL = ±2
8. DPY $\geqslant$ P for EPQ run
9. $n < 1$
10. CPU $\leqslant$ 0

BD-08

Example: A company wishes to find the order quantity and reorder point that will yield the lowest annual cost for a special type of bolt that is used on its production line. The bolt supplier furnishes the following price list:

Order Quantity	Unit Price
0 – 999	$.050
1000 – 1999	.048
2000 – 4999	.045
5000 up	.040

The annual holding costs are 15% of the inventory value. Each order placed costs the company $10. Demand per year averages 12,000 for these bolts. The expected demand during lead time is 1,000 with a standard deviation of 56. If the company wants the probability of stocking out to be no greater than 15%, find the order quantity and reorder point that yields the lowest cost.

From the information given the input values for the program are:

$n = 4$	UHC = .15
$Q_1 = 0$	CPO = 10
$CPU_1 = .05$	DPY = 12000
$Q_2 = 1000$	EDDLT = 1000
$CPU_2 = .048$	$\sigma = 56$
$Q_3 = 2000$	Control = 2
$CPU_3 = .045$	Pr = .15
$Q_4 = 5000$	
$CPU_4 = .04$	

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 08		Select Program	
	[SBR] [CLR]	0.	Initialize	
4	[2nd] [Op] 17	159.39	Repartition TI-58 only	
4	[B]	0.	Price Breaks	
0	[C]	0.	1st Quantity	
.05	[R/S]	0.	1st Price	
1000	[C]	0.	2nd Quantity	
.048	[R/S]	0.	2nd Price	
2000	[C]	0.	3rd Quantity	
.045	[R/S]	0.	3rd Price	
5000	[C]	0.	4th Quantity	
.04	[R/S]	0.	4th Price	
.15	[D]	0.15	UHC	
10	[E]	10.	CPO	
12000	[2nd] [A']	12000.	DPY	
1000	[2nd] [B']	1000.	EDDLT	
56	[2nd] [C']	56.	σ	
2	[A]	0.	EOQ w/Pr	

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
.15	[R/S]	0.15	Pr	
	[2nd] [E']	518.47	TAC	518.47
	[R/S]	6354.	EOQ	6354.
	[R/S]	1058.	R	1058.
	[R/S]	4.	E(DDLT>R)	4.
	[R/S]	0.1500	Pr	0.1500
	[R/S]	0.0212	St	0.0212

Lowest annual cost is achieved with an order quantity of 6354 units and the reorder point at 1058 units.

Register Contents

R_{00}	Used	R_{05}	Used	R_{10}	E(DDLT>R)	R_{15}	Pr	R_{20}	n	R_{25}	σ
R_{01}	Used	R_{06}	TAC[1]	R_{11}	St	R_{16}	E(DDLT>R)	R_{21}	$\epsilon = .5$	R_{26}	U
R_{02}	Used	R_{07}	EOQ/EPQ	R_{12}	TAC[2]	R_{17}	St	R_{22}	CPO	R_{27}	P
R_{03}	Used	R_{08}	R	R_{13}	EOQ/EPQ	R_{18}	UHC (actual)	R_{23}	DPY	R_{28}	CTL
R_{04}	Used	R_{09}	Pr	R_{14}	R	R_{19}	UHC (modified)	R_{24}	EDDLT	R_{29}	See Note 4

NOTES:
1. Values for minimum TAC in $R_{06}-R_{11}$
2. Values for last computed TAC in $R_{12}-R_{17}$
3. Modified value for EPQ calculation
4. Q_i and CPU_i are stored in pairs of successive registers beginning with R_{29}.

REORDER TIMING

Many businesses use a periodic review inventory system, where inventory status is reviewed in detail at stipulated intervals. How the periodic review is conducted varies from system to system. In a system where actual transactions are recorded, a periodic review may involve simply compiling the data already stored in a form that allows inventory status to be determined. However, in many situations actual inventory counts are made each period to get accurate information on present inventory levels. This program is designed as an aid in determining the timing of periodic reviews to reduce total cost.

Many costs are required as inputs to this program. To help you become familiar with these various inputs, we have listed the input variables in the order they're required by the program, along with a short explanation of the terms. They will be discussed more fully in subsequent paragraphs.

- T — Review period in years. This is the time interval between inventory reviews.
- Pr — Desired probability of a stockout in any period.
- Cb — Cost of issuing a backorder.
- t — Lead time in years. Lead time is the time interval between placing an order and receiving that order.
- C_h — Annual cost of holding inventory, expressed as a percentage of the unit value of the item.
- C_r — Cost to perform a review.
- C_p — Cost to place an order.
- D — Demand or usage rate per year of the item.
- μ — Expected demand during lead time.
- σ — Standard deviation of demand during lead time.
- P — Unit price or value of an item.

To use this program, you'll need to input several costs that may require some estimation on your part, unless they happen to be carried in your firm's accounting system. However, since minimum total cost is usually insensitive to slight variations in the cost estimates, minor errors in estimating the costs are not serious. You can determine if the model is sensitive for your inputs by varying the input data. If small variations in any one input produce large variations in the output, then the model is sensitive for that data and you will need to estimate carefully in that area. If small input variations produce small output variations, then the model is insensitive to your data.

To run the program you need to have either the cost of a backorder (Cb) or the desired probability of a backorder (Pr). The cost of a backorder is simply the sum of all the direct costs involved in issuing a backorder. For this model, we will assume that backorders are incurred in very small quantities and that the cost of each backorder is independent of the length of time for which the backorder exists. We also assume in this model that if the system is out of stock, the customer waits until the backordered item is received and the sale is not lost. If Cb is difficult to obtain you may wish to enter the probability of a backorder, Pr. You can determine Pr by subtracting the desired service level from 1. The desired service level is simply that decimal percentage of time that you want to have an item in stock. So, if you want to have the item in stock 95% of the time, service level is 0.95 and Pr = 0.05 (1 – 0.95 = 0.05).

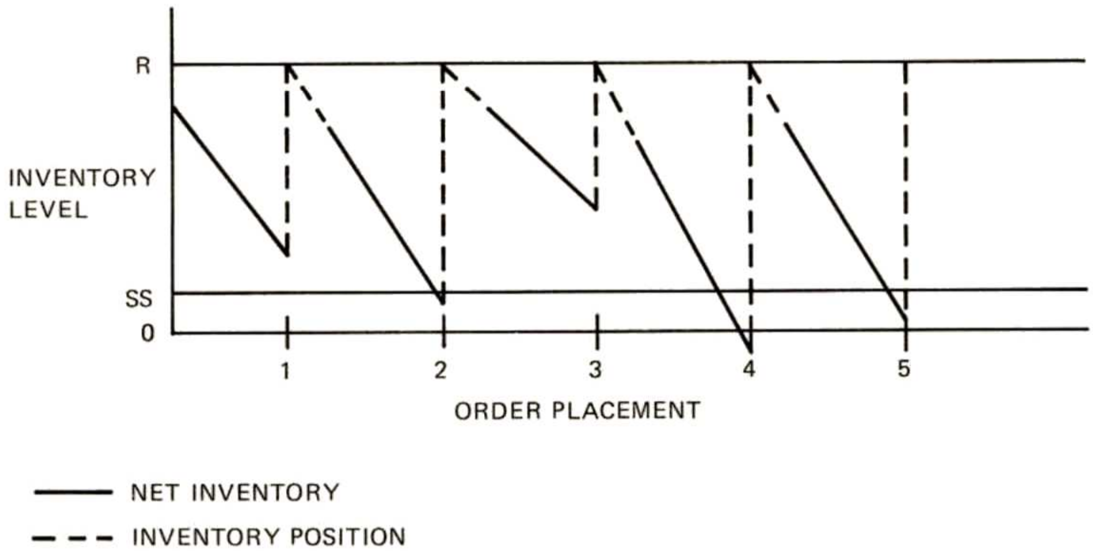
The other costs necessary to run the program may also be difficult to obtain. The cost of holding inventory (C_h) includes such costs as insurance, taxes, renting and maintaining storage facilities, and costs for obsolete and spoiled inventory. In addition the most important holding cost would be the opportunity cost. This is the cost the firm incurs by having capital invested in inventory, and is equal to the amount that could be earned if the money were invested in something other than inventory. The cost to place an order (C_p) is usually the cost of processing the order through your firm's purchasing and accounting departments.

The cost of performing a review (C_r) may vary with each review so you may need to estimate an average cost. The review cost will not include the cost of placing an order but should include all administrative and labor costs incurred as a result of making a review. The only other cost needed is the unit price (P). The unit price may include such costs as transportation costs, receiving costs, and inspection costs. Note that it is assumed in this program that the unit price is constant regardless of the quantity ordered.

The program computes many values that will be helpful in analyzing your inventory situation. For the given review period (T) and lead time (t) that you have chosen, the program will compute the total annual cost (TC) of your inventory system as well as the optimum inventory position immediately after placing an order (R). Inventory position is the sum of the number of units on hand and the number of units on order. Thus after each review you should order the number of units required to bring the inventory position up to R units.

In addition the program will compute the safety stock (SS). Safety stock is the extra inventory held as protection against the possibility of a stockout, and is the expected number of units in inventory immediately prior to the arrival of an order.

The relationship of safety stock, backorders, inventory position to net inventory is shown in the following diagram.



BD-09

Annual costs that are important to inventory control are also computed. The program will find the expected number of backorders per year (Eb) and the expected backorder costs per year (EBC). These two values may be especially important if backorders are very undesirable in your inventory system. In addition the program computes the expected review cost per year (ERC) and the expected cost per year for placing orders (EPC). These costs are based on the number of reviews conducted per year. The expected holding cost per year (EHC) is also computed. The cost of carrying the safety stock in inventory is included in EHC.

Working with this program you can try many different possible review periods (T) to determine the lowest total cost for ordering and carrying your inventory. In addition you will obtain many other values that are helpful in maintaining good inventory management

Reference:

Hadley, G. and Whitin, T.M., *Analysis of Inventory Systems,* Prentice-Hall, 1963, pp. 237–245.

Solid State Software				TI © 1978
REORDER TIMING				BD-09
D	μ	σ	P	Compute (D)
CTRL;Cb,Pr	t	C_h ; C_r ; C_p	T	Compute (P)

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 09	
2	Initialize		[SBR] [CLR]	0.
3	Enter Review Period (yrs)	T	[D]	T
4	Specify Type of Run (CTRL)	0 or 1	[A]	0.
5a	If CTRL = 0, enter prob. of backorder[1]	Pr	[R/S]	Pr
5b	If CTRL = 1, enter cost of backorder	Cb	[R/S]	Cb
6	Enter Lead Time (yrs)	t	[B]	t
7	Enter Holding Cost	C_h	[C]	C_h
8	Enter Review Cost	C_r	[R/S]	C_r
9	Enter Cost to Place an Order	C_p	[R/S]	C_p
10	Enter Demand Per Year	D	[2nd] [A']	D
11	Enter Demand During Lead Time	μ	[2nd] [B']	μ
12	Enter Std. Dev. of Demand During Lead Time	σ	[2nd] [C']	σ
13	Enter Unit Price of Item	P	[2nd] [D']	P
14a	Compute (without printer)		[2nd] [E']	TC[2]
14b	Compute (with printer)		[E]	TC R T Pr Cb Eb SS ERC EPC EHC EBC

NOTES: 1. $0 < Pr < 1$

2. Press [R/S] to obtain subsequent output values without printer.

Register Contents

R_{00} Used	R_{05} Used	R_{10} ERC	R_{15} Pr	R_{20} μ	R_{25} σ'
R_{01} Used	R_{06} SS	R_{11} t	R_{16} Eb	R_{21} σ	R_{26} Used
R_{02} Used	R_{07} EBC	R_{12} TC	R_{17} Cb	R_{22} Cp	R_{27} P
R_{03} Used	R_{08} EHC	R_{13} T	R_{18} C_h	R_{23} D	R_{28} CTRL
R_{04}	R_{09} EPC	R_{14} R	R_{19} C_r	R_{24} μ'	R_{29}

BD-09

Example: A company reviews its inventory at the end of each quarter and places orders for new stock only at that time because extra personnel are hired to do most of the work. The company has decided that it would like to have items in stock 95% of the time. The lead time for receiving orders is 4 months (1/3 year). The company usually sells about 600 units per year. Since the product sells at a constant rate throughout the year (it is not a seasonal product), the expected demand during lead time is 200 units (1/3 × 600 = 200) with a standard deviation of 21.32. It costs the company $50 to perform a review and $10 to place an order. The unit price is $15 and the annual cost of holding inventory is 20% of the unit value. Find the optimum R (inventory position immediately after placing an order) under these conditions. What is the total annual cost of the inventory system for this product? From this information, the input values are:

T = 0.25	C_p = 10
CTRL = 0	D = 600
Pr = (1 − .95) = 0.05	μ = 200
t = 1 ÷ 3	σ = 21.32
C_h = 0.2	P = 15
C_r = 50	

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 09		Select Program	
	[SBR] [CLR]	0.	Initialize	
.25	[D]	0.25	Review Period	
0	[A]	0.	Control w/probability of backorder	
.05	[R/S]	0.05	Prob. of backorder	
3	[1/X] [B]	.3333333333	Lead Time	
.2	[C]	0.2	Holding Cost	
50	[R/S]	50.	Review Cost	
10	[R/S]	10.	Order Cost	
600	[2nd] [A']	600.	Annual Demand	
200	[2nd] [B']	200.	EDDLT	
21.32	[2nd] [C']	21.32	Standard Deviation	
15	[2nd] [D']	15.	Unit Price	
	[E]		TC	639.43
			R	396.
			T	0.2500
			Pr	0.0500
			Cb	15.0000
			Eb	2.
			SS	46.
			ERC	200.
			EPC	40.
			EHC	364.
			EBC	35.22

DEMAND FORECASTING

Every businessman faces the problem of predicting the future through the various forecasts that are part of the business routine. This program is commonly used in forecasting sales. Accurate forecasting of sales is important in financial and production planning since requirements for purchases, production, and labor depend on the prediction of sales. Of the several common methods available for forecasting sales, this program uses the exponential smoothing technique. Remember that this method assumes that the future is predicted by the past, which is not always the case. However, this program along with your management judgment can be very valuable in helping you predict future sales.

This program allows up to three levels of exponential smoothing. You can use one, two, or three levels, depending upon the nature of your historical data. The different levels of smoothing are discussed in the following paragraphs, beginning with the simplest model with one level of smoothing. If you decide to use one of the more complex models, you should read through the simple models first because the information covered there is not repeated in the discussion of the more complex models.

Level Coefficient Model

This is the simplest of the exponential smoothing models and should be used only if there is no significant trend and/or seasonality in your sales (demand) data. This model uses the weighted average of the forecasted sales for the current period and the actual sales for the current period to compute the forecasted sales for the next period. The program will compute an initial value for the level coefficient (F_t) by averaging the historical sales data (Initialize Function). Once the initial value for F_t is found, you can forecast for every period in the future. The forecasted value may be updated every period as new demand data becomes available (Revise and Forecast Functions).

To run the program using this model, you need an estimate of the level smoothing constant (α_F). This constant is a value between 0 and 1 that serves as the weighting factor. The value of α_F determines how much the current historical information influences the forecast value. The closer α_F is to 1, the greater the weight of the most recent period in the forecast. A weight of $1 - \alpha_F$ is given to preceding periods. This means that the sensitivity of the forecasts to fluctuations in demand data is determined by α_F. For example, if $\alpha_F = 1$, the forecast will equal the current period's demand. If $\alpha_F = 0$, the forecast for the next period will be the same as the forecast for the current period. If you want the exponentially smoothed forecasts from this model to be equivalent to forecasts that would be obtained by using a moving average with n observations, let $\alpha_F = 2/(n + 1)$.

You can estimate α_F for your data by using the program to predict your historical data. You can vary α_F until you find a solution which closely predicts your historical data. If you find that the value required is less than zero or greater than 1, you have a trend in your data and you should use the level and trend coefficient model.

Level and Trend Coefficient Model

This model is more complex than the Level Coefficient model but uses the same basic ideas. This model is applicable if there is an upward or downward trend in your demand data, but no seasonality in the data. This model assumes a linear trend. The program will compute an initial value for the level coefficient (F_t) and for the trend coefficient (T_t) from past demand data (Initialize Function). Once the initial values are computed, you can revise the values of F_t and T_t when new demand data becomes available (Revise Function), and use the revised values for new forecasts (Forecast Function).

To use this model, you need an estimate of the level smoothing constant, α_F (explained earlier) and an estimate of the trend smoothing constant, α_T, for the trend coefficient. The trend smoothing constant, like the level smoothing constant, is a value between 0 and 1. The value of α_T determines how fast the expected demand is increasing or decreasing. Similar to α_F, a larger value of α_T increases the sensitivity of the forecasts to changes in the demand data. Thus, using two levels of smoothing, this model allows you to predict fluctuations around the actual trend line and to correct for long-term changes in the trend line.

As in the level coefficient model, you can estimate α_F and α_T by using the program to predict your historical data. You can vary α_F and α_T until you find forecasted values that best approximate your data. If you cannot find an α_F, α_T pair that yields good forecasts, you may need to use one of the models that allows for seasonality in the data.

Level and Seasonal Coefficient Model

This model is similar to the level and trend coefficient model and also has two levels of smoothing. If you have seasonality in your demand data but no trend, you should use this model. The program will compute an initial value for the level coefficient (F_t) and a seasonal coefficient for each period (S_i) from past demand data (Initialize Function). When new demand data for a particular period becomes available, the program can be used to revise the values of F_t and S_i associated with that period (Revise Function). The new values can then be used to make new forecasts (Forecast Function).

To use the model you need an estimate of the level smoothing constant, α_F (explained earlier) and an estimate of the seasonal smoothing constant for the seasonal coefficients. The seasonal constant is also a value between 0 and 1, and serves as the weighting factor for seasonal adjustments. This model uses a multiplicative, or ratio, seasonal model since the seasonal pattern is usually proportional to the demand level. As in the level coefficient model, increasing α_S makes the model more sensitive to fluctuations in the most recent seasonal data.

You can estimate α_F and α_S by using the program to predict your historical data. Note that trend in the data could cause the forecasts to become erratic. Therefore, if you cannot adequately predict your historical data with this model, you should use the level, trend, and seasonal coefficient model.

Level, Trend, and Seasonal Coefficient Model

This, the most complex model should be used when there is both trend and seasonality in the data. Using this model, an initial value can be computed for the level coefficient (F_t), the trend coefficient (T_t), and each seasonal coefficient (S_i) from past demand data. As in the level and seasonal coefficient model, there is a seasonal coefficient for each period. With the initial values, you can forecast for any period in the future (Forecast Function). When new demand data becomes available, the program can be used to revise the values of F_t, T_t and the S_i associated with that period (Revise Function). With the new values, you can make new forecasts (Forecast Function).

To run the program using this model, you need values of the level constant (α_F), the trend constant (α_T) and the seasonal constant (α_S). These were all explained in the previous models. As in the other models, you can try varying the values until you can predict your historical data. Typically, small values are used for the constants (usually 0.15) so that long-range effects are reflected in the forecasts. When using this model, you must have the old demand value and the new demand value for the first period in order to use the initialize function.

References:

Winters, Peter R., "Forecasting Sales by Exponentially Weighted Moving Averages," *Management Science,* Vol. 6, (1960), pp. 324–342.

Gross and Peterson, *Business Forecasting,* Houghton Mifflin Co., 1976, pp. 142–157.

Wheelwright and Makpidakis, *Forecasting Methods for Management,* Wiley-Interscience Publications, 1973, pp. 48–62.

Solid State Software			TI ©1978	
DEMANDING FORECASTING				BD-10
$F_t; T_t$	Card In	$\tau; D_\tau$	CTRL	Compute (D)
$\alpha_F; \alpha_T; \alpha_S$	M; S_i	t; D_i	Limit	Compute (P)

USER INSTRUCTIONS

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
	INITIALIZE FUNCTION			
1	Select Program		[2nd] [Pgm] 10	
2	Initialize		[SBR] [CLR]	0.
3	Enter Level Constant	α_F	[A]	α_F
4	Enter Trend Constant[1]	α_T	[R/S]	α_T
5	Enter Seasonal Constant[1]	α_S	[R/S]	α_S
6	Enter number of time periods of historical data[6]	t	[C]	0.
7	Enter Historical Data	D_i	[R/S]	0.
	Repeat until t values have been entered.			
8	Enter τ = 0 if t + 1 periods of historical data required[2]	0	[2nd] [C']	0.
9	Enter data for period (t + 1)	D_{t+1}	[R/S]	D_{t+1}
10	CTRL = −1 for Initialize	1	[+/−] [2nd] [D']	−1.
11	Compute[3,4]		[E]	T_t* F_t* S_i; i = 1, M*
	REVISE FUNCTION			
1	Select Program		[2nd] [Pgm] 10	
2	Initialize		[SBR] [CLR]	0.
3	Enter Level Constant[8]	α_F	[A]	α_F
4	Enter Trend Constant	α_T	[R/S]	α_T
5	Enter Seasonal Constant	α_S	[R/S]	α_S
6	Enter number of time periods of seasonal periodicity[5]	M	[B]	0.
7	Enter seasonal coefficient[5]	S_i	[R/S]	0.
	Repeat Step 7 until M values have been entered			
8	Enter Level Coefficient	F_t	[2nd] [A']	F_t
9	Enter Trend Coefficient	T_t	[R/S]	T_t
10	Enter time period of new actual demand value	τ	[2nd] [C']	0.
11	Enter new actual value	D_τ	[R/S]	D_τ
12	Set CTRL = 1 for Revise	1	[2nd] [D']	1.
13	Compute[3,4]		[E]	F_τ* T_τ* S_τ*

STEP	PROCEDURE	ENTER	PRESS	DISPLAY
	FORECAST FUNCTION			
	Steps 1 through 9 same as REVISE			
10	Enter time period of latest actual data	τ	[2nd] [C′]	0.
11	Enter number of time periods to be forecasted[7]	Limit	[D]	Limit
12	Set CTRL = 0 for Forecast	0	[2nd] [D′]	0.
13	Compute[3]		[E]	i* $DF_{t,\,t+1}$* (i = 1, limit)

NOTES:

1. If no trend and/or seasonal components in data, α_T and/or α_S = 0.
2. t + 1 periods of historical data are required if data contains *both* trend and seasonal components.
3. Key [E] is used in conjunction with PC-100A. [2nd] [E′] should be used in the absence of a PC-100A.
4. With the TI-59: α_F, α_T, α_S, M, S_i (i = 1, M), F_t, and T_t may be stored on a magnetic card by pressing [2nd] [Fix] 9, 3 [2nd] [Write], insert card in slot.
5. M and S_i required only if $\alpha_S \neq 0$.
6. t must equal M (seasonal periodicity) if $\alpha_S \neq 0$.
7. Limit must be ≥1.
8. For the TI-59, Steps 3 through 9 may be replaced by: [2nd] [B′], insert data card in slot.

* Printed on PC-100A.

Register Contents

R_{00} Used	R_{10} ΣD_i	R_{20}	R_{30} D_1	R_{40} D_{11}	R_{50} S_6
R_{01} Used	R_{11}	R_{21}	R_{31} D_2	R_{41} D_{12}	R_{51} S_7
R_{02} Used	R_{12}	R_{22}	R_{32} D_3	R_{42} α_F	R_{52} S_8
R_{03} Used	R_{13}	R_{23}	R_{33} D_4	R_{43} α_T	R_{53} S_9
R_{04} Used	R_{14}	R_{24}	R_{34} D_5	R_{44} α_S	R_{54} S_{10}
R_{05} t	R_{15}	R_{25}	R_{35} D_6	R_{45} S_1	R_{55} S_{11}
R_{06} τ	R_{16}	R_{26}	R_{36} D_7	R_{46} S_2	R_{56} S_{12}
R_{07} Used	R_{17}	R_{27}	R_{37} D_8	R_{47} S_3	R_{57} M
R_{08} Limit	R_{18}	R_{28}	R_{38} D_9	R_{48} S_4	R_{58} F_t
R_{09} CTRL	R_{19}	R_{29} D_{t+1}	R_{39} D_{10}	R_{49} S_5	R_{59} T_t

BD-10

Example: You are setting up a formal forecasting system for your business at the end of January 1978. Because you suspect there are seasonal and trend influences in the demand data that you compiled over the previous year, you choose to apply all three constants to the forecast model. The value of 0.15 is a typical value for the smoothing constants for the majority of cases. Your records for 1977 show actual demand was:

Jan.	115	Apr.	137	July	154	Oct.	138
Feb.	124	May	135	Aug.	144	Nov.	122
Mar.	122	June	145	Sept.	139	Dec.	135

Demand for Jan. 1978 was 118. From this data, initialize the forecasting program and forecast the demand for Feb. and Mar. 1978 as follows:

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 10		Select Program	
	[SBR] [CLR]	0.	Initialize	
.15	[A]	0.15	Enter α_F	
.15	[R/S]	0.15	Enter α_T	
.15	[R/S]	0.15	Enter α_S	
12	[C]	0.	Enter t	
115	[R/S]	0.	Enter D_1	
124	[R/S]	0.	Enter D_2	
122	[R/S]	0.	Enter D_3	
137	[R/S]	0.	Enter D_4	
135	[R/S]	0.	Enter D_5	
145	[R/S]	0.	Enter D_6	
154	[R/S]	0.	Enter D_7	
144	[R/S]	0.	Enter D_8	
139	[R/S]	0.	Enter D_9	
138	[R/S]	0.	Enter D_{10}	
122	[R/S]	0.	Enter D_{11}	
135	[R/S]	0.	Enter D_{12}	
0	[2nd] [C']	0.	Enter T = 0	
118	[R/S]	118.	Enter New D_1	
1	[+/−] [2nd] [D']	−1.	CTRL = −1 for Initialize	
	[E]		T_t	0.25000
			F_t	135.79167
			Period 1	1.
			S_1	0.86898
			Period 2	2.
			S_2	0.93204
			Period 3	3.
			S_3	0.91529
			Period 4	4.
			S_4	1.02590
			Period 5	5.
			S_5	1.00903
			Period 6	6.
			S_6	1.08175
			Period 7	7.
			S_7	1.14676

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
			Period 8	8.
			S_8	1.07030
			Period 9	9.
			S_9	1.03122
			Period 10	10.
			S_{10}	1.02191
			Period 11	11.
			S_{11}	0.90176
			Period 12	12.
		12.00000*	S_{12}	0.99600

*With the TI-59, this data may be recorded on a magnetic card by pressing [2nd] [Fix] 9, 3 [2nd] [Write] and inserting card.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
1	[2nd] [C′]	0.00000	Period of Latest Actual Data	
2	[D]	2.00000	No. Forecast Periods	
0	[2nd] [D′]	0.00000	0 for Forecast	
	[E]		Feb.	1.
			Feb. Forecast	126.796
			Mar.	2.
		2.000	Mar. Forecast	124.746

At the end of February your records showed actual demand for this month to be 130. Revise the program data to comprehend this new input and reforecast March.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 10		Select Program	
	[SBR] [CLR]	0.	Initialize	
*.15***	[A]	0.15	Enter α_F	
.15	[R/S]	0.15	Enter α_T	
.15	[R/S]	0.15	Enter α_S	
12	[B]	0.	Enter M	
.86898	[R/S]	0.	Enter S_1	
.93204	[R/S]	0.	Enter S_2	
.91529	[R/S]	0.	Enter S_3	
1.0259	[R/S]	0.	Enter S_4	
1.00903	[R/S]	0.	Enter S_5	
1.08175	[R/S]	0.	Enter S_6	
1.14676	[R/S]	0.	Enter S_7	
1.07030	[R/S]	0.	Enter S_8	
1.03122	[R/S]	0.	Enter S_9	
1.02191	[R/S]	0.	Enter S_{10}	
0.90176	[R/S]	0.	Enter S_{11}	
.996	[R/S]	0.	Enter S_{12}	
135.79167	[2nd] [A′]	135.79167	Enter F_t	
.25	[R/S]	.25	Enter T_t	
2	[2nd] [C′]	0.	Enter τ	
130	[R/S]	130.	Enter D_τ	
1	[2nd] [D′]	1.	1 for Revise	

**If a magnetic card was used for data storage, all data in italics can be entered at this point by pressing [2nd] [B′], and inserting the card. When the data has been read properly, 3. will appear in the display.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[E]		New F_t	136.55727
			New T_t	0.32734
		0.93503*	New S_2	0.93503
2	[2nd] [C']	0.00000	Period of Latest Actual Data	
1	[D]	1.	No. Forecast Periods	
0	[2nd] [D']	0.	0 for Forecast	
	[E]		March	1.
		1.000	March Forecast	125.289

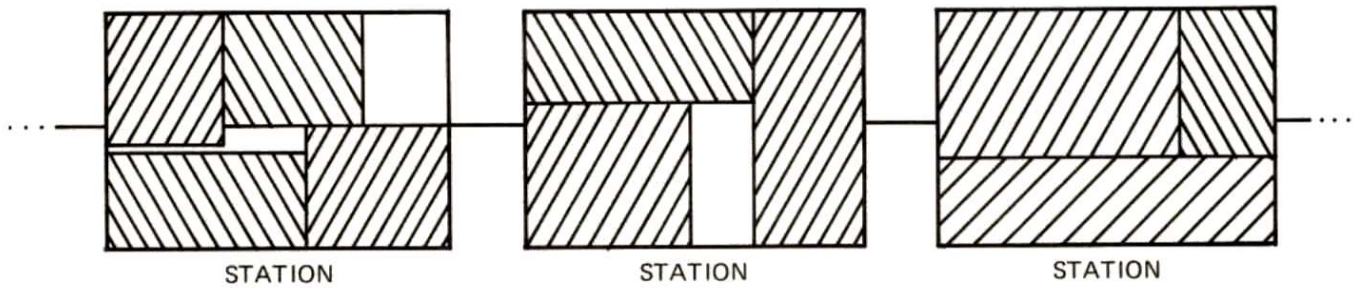
*With the TI-59, this data may be recorded on a magnetic card by pressing [2nd] [Fix] 9, 3 [2nd] [Write] and inserting card.

ASSEMBLY LINE BALANCING

In any business situation, time is money! On an assembly line where repeated operations are involved over long periods of time – the better the design of the line, the less is paid for idle time. Reducing wasted time on the assembly line obviously means lower production cost – greater profit potential. This program is designed to act as an aid in efficient assembly line design and balancing. Used effectively, this program can help you trim costly idle time from plant operations, and help ensure a smooth work flow as a product is assembled.

An assembly line is a series of *work stations*. At each work station one or more *tasks* may be performed, each of which may have a different task time. To *balance* the line you need to determine what tasks to assign to each work station to minimize *idle* (wasted) *time*. In an assembly line situation the item being assembled spends the same amount of time at each work station, called the cycle time. If there is no idle time at any work station, the line is perfectly balanced.

One way to view the problem is shown in the diagram below – where we're packing tasks of different times (represented here by area) into work stations with the same cycle time.



In our simplified diagram above, the area in each station that is not covered by a task (shaded boxes) represents wasted time. So balancing the line involves choosing a cycle time and arranging the tasks so that wasted time is minimized.

In most assembly situations, however, – there's one more "catch". Some tasks usually need to be completed before others can take place. This limits the way in which we can pack the tasks (with different *task times*) into the stations (with constant *cycle times*). This program will help you find the best way to assign tasks to work stations – taking into account the fact that some tasks must occur before others. We'll assume for this analysis that there are no restrictions on the assembly line imposed by stationary machines or other physical limitations. In other words we'll assume the tasks may be assigned to any work station.

To help keep track of which tasks must occur before others, we'll begin by drawing what is called a *precedence diagram.* This diagram will illustrate schematically the order in which the tasks must be performed. The time needed to complete each task is usually included on such diagrams.

BD-11

Let's consider how we would begin approaching a simplified example case. A car manufacturer has an assembly line that handles final dashboard assembly, carpeting and glass installation, inside and outside trim, installation of seats; as well as final cleaning and polishing before shipment to the dealer.

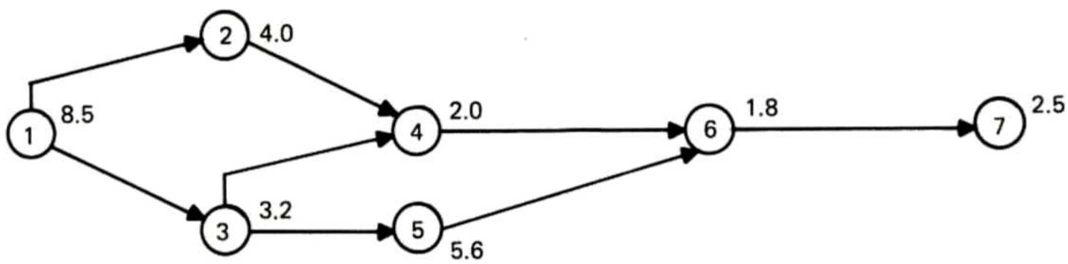
The following table lists the tasks to be completed.

Task Number	Task Name	Time Needed
1	Final dashboard assembly	8.5
2	Carpeting installation	4.0
3	Glass installation	3.2
4	Inside trim	2.0
5	Outside trim, bumpers, etc.	5.6
6	Seat installation	1.8
7	Clean and polish	2.5

There are some tasks that must be completed before others in building the car. We can build a table that will help us in drawing the precedence diagram. The table below shows for each task, those tasks which must be completed before the particular task can be started.

Task Number	Predecessors
1	
2	Dashboard (1)
3	Dashboard (1)
4	Carpet (2), Glass (3)
5	Glass (3)
6	Inside trim (4), Outside trim (5)
7	Seats (6)

The precedence diagram we now draw, represents this task-predecessor relationship. We show each task number in a circle, with the time for completion as a number above and to the right of each task:



The pathways we must allow are shown as arrows that connect the circles from task 1 to task 7. In this example the tasks are arranged by number in ascending order. This is recommended since the task with the largest task number may not have any successors.

When the number of tasks to be assigned to work stations becomes large, the number of possible arrangements becomes difficult to enumerate. Since we cannot list all the possible solutions and pick the best one, logical decision rules are usually used to arrive at satisfactory solutions. In general, we have no way of knowing if we have the best solution unless we obtain a perfect (no idle time) balance.

This program can help you quickly survey alternatives and get to a good solution for your particular line situation. You can select and try out different *cycle times* for your line, and the program will follow logical decision rules in assigning tasks for you. For each cycle time you choose the program determines the number of work stations required and computes the idle time at each work station.

Balance delay is the amount of idle time on the line (measured as a percentage). You can compute it with the following equation:

$$\text{Balance delay} = \frac{\text{Total Idle Time}}{\text{(Number of Stations) (Cycle Time)}} \times 100$$

For any line, the cycle time and task arrangement giving the smallest balance delay is the best solution.

In assigning tasks to work stations, certain decision rules are used in this program. The first decision procedure finds out if a task is available. A task is available if its immediate predecessors are all assigned to work stations. From those tasks that are available, tasks are chosen that will fit in a work station. A task will fit in a work station if the task time is less than the remaining station time. (Remaining station time is determined by subtracting the total time for the tasks assigned to a work station from the cycle time.) From the tasks that will fit in a work station, the task with the largest *factor* (each task's *factor* is computed by multiplying the task time by 1 plus the number of the task's immediate successors) is assigned to that work station. If two or more tasks have the same factor, then the task with the lowest task number is assigned to the work station. When a task is assigned to a work station, the task time is subtracted from the old remaining station time to obtain a new remaining station time. This process is repeated until no available task will fit in the work station. We begin again with a new work station and the cycle time becomes the remaining station time. When all the tasks have been assigned to work stations, you can compute the total idle time (the sum of the idle times for all stations) and the balance delay.

To give you a picture of the process, we'll mentally go through the task assignments for our assembly line example. For a first try we'll select a cycle time of 8.5 hours. (Note that you cannot choose a cycle time that is less than your shortest task time.) To begin, the only task with no predecessors is task 1 so it is assigned to station 1. The remaining station time is 0.0. Obviously, no other task will fit in this station. For station 2 both tasks 2 and 3 are available since they now have no unassigned predecessors. Both tasks will also fit in this work station. Task 3 (Factor = 9.6) is assigned first since it has a larger factor than task 2 (Factor = 8.0). The new remaining station time is 5.3 (8.5 – 3.2 = 5.3). Now the available tasks are task 2 and task 5. Task 5 will not fit in station 2 since the task time is 5.6. Since the task time for task 2 is 4.0, task 2 will fit in station 2. The new remaining station time is 1.3 (5.3 – 4.0 = 1.3). Now task 4 and task 5 become available. However neither of these two will fit in station 2. Therefore, the idle time at station 2 is 1.3. Both task 4 and task 5 will fit in station 3. Task 5 (Factor = 11.2) is selected first since it has a larger factor than task 4 (Factor = 4.0). After task 5 is assigned to station 3 the remaining station time is 2.9 (8.5 – 5.6 = 2.9). Now task 4 is the only available task. It will fit in station 3 so it is assigned. The new remaining station time is 0.9 (2.9 – 2.0 = 0.9). Next, task 6 becomes available. It will not fit in station 3 so the idle time at station 3 is 0.9. In the same manner task 6 and then task 7 are assigned to station 4. The idle time at station 4 is 4.2 [8.5 – (1.8 + 2.5) = 4.2]. We can now compute the balance delay.

$$\text{Balance delay} = \frac{(0.0 + 1.3 + 0.9 + 4.2)}{(4)\ (8.5)} \times 100 = 18.8\%$$

If we try different cycle times we may find a better solution.

Using your calculator you can try many different solutions by using different cycle times. While it is difficult to be sure you've reached the *absolute best* cycle time, you can work to minimize your balance delay and come pretty close. If, however, you do not get satisfactory results with this program, you may want to try some different decision rules on your own. The decision rules used in this program have been shown to give good solutions most of the time. Thus, with the help of this program you may find many satisfactory solutions and then your decisions may be made on other appropriate factors.

References:

Arcus, Albert L., "COMSOAL: A Computer Method of Sequencing Operations for Assembly Lines," *The International Journal of Production Research*, Vol. 4, No. 4, 1966, pp. 259-277.

Mansoor, E.M. Dar-El., "MALB-A Heuristic Technique for Balancing Large Single-Model Assembly Lines," *AIIE Transactions*, December 1973, pp. 343-356.

Solid State Software				TI ©1978
ASSEMBLY LINE BALANCING				BD-11
				Cycle T → (D)
No. Tasks	Task Time	Followers	Entry Comp.	Cycle T → (P)

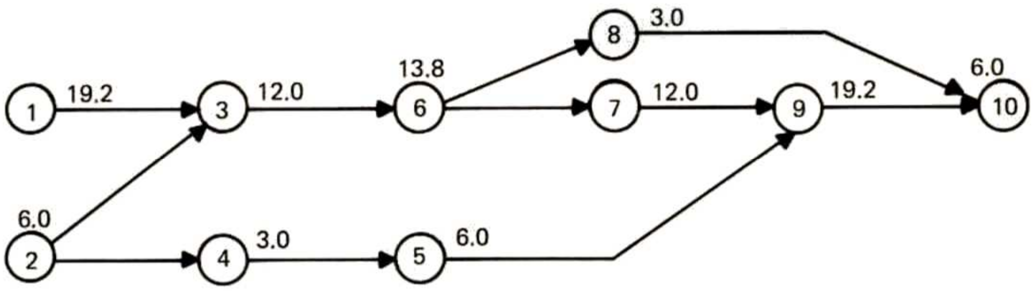
STEP	PROCEDURE	ENTER	PRESS	DISPLAY
1	Select Program		[2nd] [Pgm] 11	
2	Initialize		[SBR] [CLR]	0.
3	Enter Number of Tasks	No. Tasks	[A]	No. Tasks
4	Enter Task Time of Task i[1] (in XXX.X format)	Task Time	[B]	Task Time
5	Enter Task Number of Each Follower of Task i (Repeat as Needed)	Follower No.	[C]	Follower No.
6	Repeat Steps 4 and 5 for each Task			
7	After all Entries are made		[D]	0.
8a	Enter Cycle Time (with printer)	Cycle Time[2]	[E]	See Note 3
8b	Enter Cycle Time (without printer)	Cycle Time[2]	[2nd] [E']	See Note 4

NOTES:

1. The format of the task time is restricted to 4 digits, 3 to the left of the decimal and 1 to the right. Tasks must be entered in numerical sequence starting with Task No. 1.
2. The cycle time must be greater than or equal to the largest task time. If the cycle time is too small, the largest task time is flashed in the display.
3. The cycle time is printed upon entry. Then the number of each work station is printed followed by the numbers of the tasks assigned and the idle time at the station. When all tasks have been assigned, 0. is displayed following the last idle time.
4. Without the printer, [R/S] must be pressed between each output. Work station numbers are displayed as negative numbers. The numbers of the tasks assigned to the station are then displayed one at a time. Finally the idle time is flashed in the display. When all tasks have been assigned, 0. is displayed following the last idle time.

BD-11

Example: An assembly line consists of 10 tasks having the task times shown in the following precedence diagram. Evaluate the balance of the line using cycle times of 19.2 and 25.8.



ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT	
	[2nd] [Pgm] 11		Select Program		
	[SBR] [CLR]	0.	Initialize		
10	[A]	10.	No. Tasks		
19.2	[B]	19.2	Task 1	1.	TASK
				19.2	
3	[C]	3.	Follower	3.	
6	[B]	6.0	Task 2	2.	TASK
				6.0	
3	[C]	3.	Follower	3.	
4	[C]	4.	Follower	4.	
12	[B]	12.0	Task 3	3.	TASK
				12.0	
6	[C]	6.	Follower	6.	
3	[B]	3.0	Task 4	4.	TASK
				3.0	
5	[C]	5.	Follower	5.	
6	[B]	6.0	Task 5	5.	TASK
				6.0	
9	[C]	9.	Follower	9.	
13.8	[B]	13.8	Task 6	6.	TASK
				13.8	
7	[C]	7.	Follower	7.	
8	[C]	8.	Follower	8.	
12	[B]	12.0	Task 7	7.	TASK
				12.0	
9	[C]	9.	Follower	9.	
3	[B]	3.0	Task 8	8.	TASK
				3.0	
10	[C]	10.	Follower	10.	
19.2	[B]	19.2	Task 9	9.	TASK
				19.2	
10	[C]	10.	Follower	10.	
6	[B]	6.0	Task 10	10.	TASK
				6.0	
	[D]	0.	Entry Complete		

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT	
19.2	[E]		Cycle Time	19.2	
			Work Station No. 1	1.	STA
			Task No.	1.	
			Idle Time	0.0	
			Work Station No. 2	2.	STA
				2.	
				3.	
				1.2	
			Work Station No. 3	3.	STA
				6.	
				4.	
				2.4	
			Work Station No. 4	4.	STA
				7.	
				5.	
				1.2	
			Work Station No. 5	5.	STA
				9.	
				0.0	
			Work Station No. 6	6.	STA
				8.	
				10.	
		0.		10.2	
25.8	[E]		Cycle Time	25.8	
			Work Station No. 1	1.	STA
				1.	
				2.	
				0.6	
			Work Station No. 2	2.	STA
				3.	
				6.	
				0.0	
			Work Station No. 3	3.	STA
				7.	
				4.	
				5.	
				8.	
				1.8	
			Work Station No. 4	4.	STA
				9.	
				10.	
		0.		0.6	

A cycle time of 19.2 results in 6 work stations with idle times of 0.0, 1.2, 2.4, 1.2, 0.0, and 10.2 for a total idle time of 15 with total time of 6 X 19.2 = 115.2. This yields a balance delay of 13%.

A cycle time of 25.8 results in 4 work stations with idle times of 0.6, 0.0, 1.8, and 0.6 for a total idle time of 3 with total time of 103.2. This yields a balance delay of 2.9%.

Register Contents

R_{00} Used	R_{05} Used	R_{10} See Note	R_{15}	R_{20}	R_{25}
R_{01} Pointer	R_{06} Max. Task T	R_{11}	R_{16}	R_{21}	R_{26}
R_{02} No. Tasks	R_{07} Used	R_{12}	R_{17}	R_{22}	R_{27}
R_{03} Pointer	R_{08} Used	R_{13}	R_{18}	R_{23}	R_{28}
R_{04} Counter	R_{09} Used	R_{14}	R_{19}	R_{24}	R_{29}

NOTE: Registers R_{10} through R_{9+N} (where N is the number of tasks) are used to store individual task information in the format:

$$\text{Task Time} \times 10.\ \underbrace{\text{\# Followers}}_{\text{2 digits}}\ \underbrace{\text{\# Predecessors}}_{\text{2 digits}}\ \underbrace{\text{String Pointer}}_{\text{2 digits}}$$

R_{10+N} and above are used to store predecessor relationships in string format, packed 5 to a register (i.e., the follower numbers of each task are located here – the string pointer tells the program where to look for the information and the # followers tells it how far to go).

The number of registers required is:

10 + # Tasks + INT [(# Arrows/5) + .9]

where INT means you should discard the fractional part of the number in brackets.

SHORT-TERM FINANCING REQUIREMENTS

Short-term cash flow forecasting is required in many business situations. The Project Planning and Budgeting Program (BD-05) can be used to advantage in preparing such forecasts. The following example shows how a typical problem might be attacked using the program. Two basic assumptions are made in this example. First, the TI-58 calculator is used, limiting us to a maximum of four rows. Second, use of the PC-100A printer is assumed.

The situation is: Your company wants to have, at month end, a working capital balance equal to twice that month's sales. Based on a number of factors, you must project how much money will have to be borrowed each month to meet this requirement. Sales are projected to grow at 2% per month with January sales set at $10 million. Receipts follow sales by two months. Payroll for January is $5 million growing at 2% per month. Cost of component parts are typically 25% of sales and payments follow sales by one month. Other costs (in $ million) are projected to be:

Jan.	$1.62	Apr.	$1.69	Jul.	$1.71	Oct.	$2.52
Feb.	1.90	May	1.76	Aug.	1.48	Nov.	2.24
Mar.	1.15	Jun.	2.15	Sep.	2.20	Dec.	1.70

The year-end balance in working capital was $17.90 million.

Before starting data entry, detailed definition of the steps to be performed can avoid time-consuming and frustrating errors.

Step 1: Enter Sales in Row 1 as 10.00 ($ million) growing at 2% per month.

Step 2: Receipts follow Sales by two months. Enter Receipts in Row 2 as Row 1 shifted two columns to the right. Columns 1 and 2 have to be entered manually.

Col 1: $10/1.02^2 = 9.61$ Col 2: $10/1.02 = 9.80$

Step 3: Enter Payroll in Row 3 as 5.00 ($ million) growing at 2% per month.

Step 4A: Component Payments are 25% of Sales and follow Sales by one month. Begin this computation by entering Row 4 as Row 1 shifted one column to the right. Column 1 is entered manually.

Col 1: $10/1.02 = 9.80$

Step 4B: Complete computation of Component Payments by multiplying Row 4 by 0.25, and entering the result in Row 4.

Step 5: Add Row 4 to Row 3 to compute sum of Payroll and Component Payments. Enter this sum in Row 3.

Step 6: Enter Other Payments in Row 4 as individual elements.

Step 7: Add Row 4 to Row 3 to compute Total Payments. Enter this sum in Row 3.

Step 8: Multiply Row 1 (Sales) by 2 to compute the Desired Balance. Enter result in Row 1.

Step 9: Enter last month's balance in Row 4 as Row 1 (Desired Balance) shifted one column to the right. Enter 17.90 in Column 1.

Step 10: Subtract Row 3 (Total Payments) from Row 2 (Receipts) to determine Net Income. Enter in Row 3.

Step 11: Subtract Row 3 (Net Income) from Row 1 (Desired Balance) to determine money needed from other sources. Enter in Row 3.

Step 12: Subtract Row 4 (Previous Balance) from Row 3 (Money Needed) to determine amount to be borrowed each month. Enter in Row 4.

As the various numbers are calculated they can be written in tabular format (a spread sheet) for easy reference. The calculator procedure is as follows, with the results tabulated.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
	[2nd] [Pgm] 05*		Select Program	
	[SBR] [CLR]	0.00	Initialize	
4	[2nd] [B']	4.00	Enter Max. Rows	
1	[2nd] [C']	1.00	Row No.	
10	[C]	10.00	Base Sales	SEE TAPE 1
.02	[R/S]	1.00	Growth Rate	
1	[E]	134.12	Row 1 Sum	
1	[A]	1.00	Row to Shift	
2	[R/S]	2.00	Resulting Row	
2	[2nd] [D']	0.00	Shift Right 2 Col.	
9.61	[R/S]	9.61	Col. 1	
9.8	[R/S]	9.80	Col. 2	
2	[B]	2.00	Row No.	
	[R/S]	0.00	Recall Row	SEE TAPE 2
2	[E]	128.91	Row 2 Sum	
3	[2nd] [C']	3.00	Row No.	
5	[C]	5.00	Base Payroll	SEE TAPE 3
.02	[R/S]	3.00	Growth Rate	
3	[E]	67.06	Row 3 Sum	
1	[A]	1.00	Row to Shift	
4	[R/S]	4.00	Resulting Row	
1	[2nd] [D']	0.00	Shift Right 1 Col.	
9.8	[R/S]	9.80	Col. 1	
.25	[2nd] [A']	0.25	Enter Constant	
4	[R/S]	4.00	Row No.	
	[SBR] [X]	0.00	Multiply	SEE TAPE 4
4	[E]	32.87	Row 4 Sum	
4	[A]	4.00	Comp. Pmts.	
3	[R/S]	3.00	Payroll	
	[SBR] [+]	99.93	Sum 3 + 4	SEE TAPE 5
4	[2nd] [C']	4.00	Row No.	
1.62	[C]	1.62	Col. 1	SEE TAPE 6
1.9	[C]	1.90	Col. 2	
1.15	[C]	1.15	Col. 3	
1.69	[C]	1.69	Col. 4	

*With the TI-58, press 6 [2nd] [Op] 17 to repartition before proceeding.

ENTER	PRESS	DISPLAY	COMMENTS	PRINTOUT
1.76	[C]	1.76	Col. 5	
2.15	[C]	2.15	Col. 6	
1.71	[C]	1.71	Col. 7	
1.48	[C]	1.48	Col. 8	
2.2	[C]	2.20	Col. 9	
2.52	[C]	2.52	Col. 10	
2.24	[C]	2.24	Col. 11	
1.7	[C]	1.70	Col. 12	
4	[E]	22.12	Row 4 Sum	
4	[A]	4.00	Other Pmts.	
3	[R/S]	3.00	Comp. + Payroll	
	[SBR] [+]	122.05	Total Pmts.	SEE TAPE 7
2	[2nd] [A']	2.00	Enter Constant	
1	[R/S]	1.00	Row No.	
	[SBR] [×]	268.24	Multiply	SEE TAPE 8
1	[A]	1.00	Row to Shift	
4	[R/S]	4.00	Resulting Row	
1	[2nd] [D']	0.00	Shift Right 1 Col.	
17.9	[R/S]	17.90	Col. 1	
4	[B]	4.00	Row No.	
	[R/S]	22.12	Recall Row	SEE TAPE 9
4	[E]	261.27	Row 4 Sum	
2	[A]	2.00	Receipts	
3	[R/S]	3.00	Total Pmts.	
	[SBR] [–]	6.85	Subtract	SEE TAPE 10
1	[A]	1.00	Desired Balance	
3	[R/S]	3.00	Net Income	
	[SBR] [–]	261.39	Subtract	SEE TAPE 11
3	[A]	3.00	Money Needed	
4	[R/S]	4.00	Previous Balance	
	[SBR] [–]	0.11	Subtract	SEE TAPE 12

TAPE 1		TAPE 2		TAPE 3		TAPE 4	
1.00	ROW	2.00	ROW	3.00	ROW	4.00	ROW
10.00		9.61		5.00		2.45	
10.20		9.80		5.10		2.50	
10.40		10.00		5.20		2.55	
30.60	Σ	29.41	Σ	15.30	Σ	7.50	Σ
10.61		10.20		5.31		2.60	
10.82		10.40		5.41		2.65	
11.04		10.61		5.52		2.71	
32.48	Σ	31.22	Σ	16.24	Σ	7.96	Σ
11.26		10.82		5.63		2.76	
11.49		11.04		5.74		2.82	
11.72		11.26		5.86		2.87	
34.47	Σ	33.13	Σ	17.23	Σ	8.45	Σ
11.95		11.49		5.98		2.93	
12.19		11.72		6.09		2.99	
12.43		11.95		6.22		3.05	
36.57	Σ	35.15	Σ	18.29	Σ	8.96	Σ
		0.00				0.00	
134.12	ΣR	128.91	ΣR	67.06	ΣR	32.87	ΣR

TAPE 5		TAPE 6		TAPE 7		TAPE 8	
3.00	ROW	4.00	ROW	3.00	ROW	1.00	ROW
7.45		1.62		9.07		20.00	
7.60		1.90		9.50		20.40	
7.75		1.15		8.90		20.81	
22.80	Σ	4.67	Σ	27.47	Σ	61.21	Σ
7.91		1.69		9.60		21.22	
8.07		1.76		9.83		21.65	
8.23		2.15		10.38		22.08	
24.20	Σ	5.60	Σ	29.80	Σ	64.95	Σ
8.39		1.71		10.10		22.52	
8.56		1.48		10.04		22.97	
8.73		2.20		10.93		23.43	
25.68	Σ	5.39	Σ	31.07	Σ	68.93	Σ
8.90		2.52		11.42		23.90	
9.08		2.24		11.32		24.38	
9.26		1.70		10.96		24.87	
27.25	Σ	6.46	Σ	33.71	Σ	73.15	Σ
99.93		22.12	ΣR	122.05		268.24	

TAPE 9		TAPE 10		TAPE 11		TAPE 12	
4.00	ROW	3.00	ROW	3.00	ROW	4.00	ROW
17.90		0.54		19.46		1.56	
20.00		0.30		20.10		0.10	
20.40		1.10		19.71		−0.69	
58.30	Σ	1.94	Σ	59.27	Σ	0.97	Σ
20.81		0.60		20.62		−0.19	
21.22		0.58		21.07		−0.15	
21.65		0.24		21.85		0.20	
63.68	Σ	1.42	Σ	63.54	Σ	−0.14	Σ
22.08		0.72		21.80		−0.28	
22.52		1.00		21.97		−0.55	
22.97		0.33		23.10		0.13	
67.58	Σ	2.06	Σ	66.87	Σ	−0.71	Σ
23.43		0.06		23.84		0.41	
23.90		0.39		23.99		0.08	
24.38		0.99		23.88		−0.50	
71.71	Σ	1.44	Σ	71.71	Σ	−0.01	Σ
22.12		6.85		261.39		0.11	
261.27	ΣR						

	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.	Total
Sales	10.00	10.20	10.40	10.61	10.82	11.04	11.26	11.49	11.72	11.95	12.19	12.43	134.12
Receipts	9.61	9.80	10.00	10.20	10.40	10.61	10.82	11.04	11.26	11.49	11.72	11.95	128.91
Payroll	5.00	5.10	5.20	5.31	5.41	5.52	5.63	5.74	5.86	5.98	6.09	6.22	67.06
Component Pmts.	2.45	2.50	2.55	2.60	2.65	2.71	2.76	2.82	2.87	2.93	2.99	3.05	32.87
Other Pmts.	1.62	1.90	1.15	1.69	1.76	2.15	1.71	1.48	2.20	2.52	2.24	1.70	22.12
Total Pmts.	9.07	9.50	8.90	9.60	9.83	10.38	10.10	10.04	10.93	11.42	11.32	10.96	122.05
Desired Balance	20.00	20.40	20.81	21.22	21.65	22.08	22.52	22.97	23.43	23.90	24.38	24.87	268.24
Previous Balance	17.90	20.00	20.40	20.81	21.22	21.65	22.08	22.52	22.97	23.43	23.90	24.38	261.27
Net Income	0.54	0.30	1.10	0.60	0.58	0.24	0.72	1.00	0.33	0.06	0.39	0.99	6.85
Money Needed	19.46	20.10	19.71	20.62	21.07	21.85	21.80	21.97	23.10	23.84	23.99	23.88	261.39
Borrow	1.56	0.10	–0.69	–0.19	–0.15	0.20	–0.28	–0.55	0.13	0.41	0.08	–0.50	0.11

FACILITY SCHEDULING

During recent years an increasing number of larger manufacturing firms have installed computer-based systems for scheduling operations in fabrication-assembly shops. However, scheduling problems also must be dealt with by managers of comparatively small operations. The discussion here is not intended to be an extensive treatment of scheduling theory, but rather is presented to show how a few simple rules can be applied to help you develop an analytical approach to your decision making process.

Assume for this discussion that the following information is known for each job that is logged in your shop.

- i — Identifier (serial number) of the i^{th} job (assume 1 for the first job)
- P_i — Processing time for the i^{th} job
- W_i — Weight (importance) of the i^{th} job, scaled from 10 for the most important to 1 for the least important
- D_i — Due date of the i^{th} job (time from current date)

Obviously, performing the jobs in the order of arrival (identifier order) yields a simple first-in, first-out sequence. This arrangement is often considered to be the "fairest" sequence because all customers are treated "equally." And since it is the easiest to keep track of, it is readily endorsed by many managers. However, processing the jobs in another sequence to accomplish another goal may better utilize the available facilities, resulting in improved on-time performance also. Performing the jobs in sequences which correspond to particular increasing functions of processing time, weight, and due date can accomplish the following objectives: minimize mean lateness, minimize weighted mean lateness, minimize maximum tardiness, maximize minimum lateness. What needs to be done to accomplish the objectives is to sort the jobs as described below.

Minimize mean lateness. Sort the processing times of the jobs so that the job with the shortest processing time is completed first. Negative lateness means the job was completed before the due date.

Minimize weighted mean lateness. Sort the jobs according to the processing time/weight, putting the smallest first.

Minimize maximum tardiness. Sort the due dates so that the job with the earliest due date is performed first. Tardiness is defined as lateness if the job is late, and zero if the job is completed on or before its due date.

Maximize minimum lateness. Sort according to (processing time – due date), putting the smallest first. This will result in most jobs being completed on or near their due dates. This can be important if finished jobs cannot be shipped until the due date.

The sorting required can usually be done by hand. However, if you prefer to use the calculator, a simple number sorting program is included at the end of this section. Up to 97 numbers can be sorted with the TI-59. With the TI-58, the limit is 37 numbers.

The effects of the various sequences can be seen in the following example which comprises the following eight jobs.

Job Log

i	P_i	D_i	W_i	D_i-P_i	P_i/W_i
1	4	16	5	12	.8
2	7	16	8	9	.875
3	1	8	9	7	.111
4	6	21	3	15	2.
5	3	9	2	6	1.5
6	4	25	7	21	.571
7	6	30	5	24	1.2
8	5	34	8	29	.625

First, let's examine the effect of the various sequences in terms of lateness, which is determined by subtracting the time remaining for a particular job from the cumulative sum of processing times (ΣP_i).

i Order

i	ΣP_i	Lateness
1	4	–12
2	11	–5
3	12	4
4	18	–3
5	21	12
6	25	0
7	31	1
8	36	2

P_i Order

i	P_i	ΣP_i	Lateness
3	1	1	–7
5	3	4	–5
1	4	8	–8
6	4	12	–13
8	5	17	–17
4	6	23	2
7	6	29	–1
2	7	36	20

P_i/W_i Order

i	P_i/W_i	ΣP_i	Lateness
3	.111	1	–7
6	.571	5	–20
8	.625	10	–24
1	.8	14	–2
2	.875	21	5
7	1.2	27	–3
5	1.5	30	21
4	2.	36	15

D_i Order

i	D_i	ΣP_i	Lateness
3	8	1	–7
5	9	4	–5
1	16	8	–8
2	16	15	–1
4	21	21	0
6	25	25	0
7	30	31	1
8	34	36	2

D_i – P_i Order

i	D_i–P_i	ΣP_i	Lateness
5	6	3	–6
3	7	4	–4
2	9	11	–5
1	12	15	–1
4	15	21	0
6	21	25	0
7	24	31	1
8	29	36	2

Although each of the sequences attains its primary objective, the overall effect should be examined before arbitrarily applying any of the sorting rules.

Number Sorting Program

Your calculator can be programmed to perform number sorting by performing the following operations.

1. Press 10 [2nd] [Op] 17 on the TI-59, or press 4 [2nd] [Op] 17 on the TI-58.

2. Press [RST] [LRN] and enter the following program:

   [2nd] [Lbl] [B] [2nd] [Adv]
   [RCL] 00 [STO] 99 (39 with TI-58)
   [RCL] [2nd] [Ind] 00
   [INV] [2nd] [St flg] 7
   [2nd] [Op] 18
   [2nd] [If flg] 7024
   [CLR] [2nd] [Dsz] 0007
   [INV] [SBR] [x ≷ t] [RCL] 00
   [STO] 98 (38 with TI-58)
   [INV] [2nd] [Dsz] 0061
   [RCL] [2nd] [Ind] 00
   [INV] [2nd] [St flg] 7
   [2nd] [Op] 19
   [2nd] [If flg] 7056
   [2nd] [x=t] 051
   [2nd] [x ≥ t] 056
   [x ≷ t] [RCL] 00
   [STO] 98 (38 with TI-58)
   [CLR]
   [2nd] [Dsz] 0034
   [RCL] 98 (38 with TI-58)
   [2nd] [Prt] ([R/S] without printer)
   0 [1/x]
   [STO] [2nd] [Ind] 98 (38 with TI-58)
   [CLR]

(Continued on next page)

[RCL] 99 (39 with TI-58)
[STO] 00 [GTO] 007
[2nd] [Lbl] A
[2nd] [Op] 20
[STO] [2nd] [Ind] 00
[2nd] [Prt] [RCL] 00
[INV] [SBR]
[LRN] *

3. Press [2nd] [CMs]

4. Enter data and press [A], repeat until all data is entered.

5. Press [B] to obtain results.

   Note, you must start with Step 3 each time the program is repeated.

*With the TI-59, the program can be recorded on a magnetic card by pressing 1 [2nd] [Write]. The program can be entered from the card by setting the partition, pressing [CLR] and inserting card.

APPENDIX A

APPENDIX A – EQUATIONS

BD-02 LONG-TERM FINANCING

$$P = REV \times \%GPS$$
$$D = P \times \%DR$$
$$RE = P - D$$
$$CAP = REV \times CAS$$
$$FN = CAP - TA - E$$

where

P = Gross Profit
REV = Expected revenue in sales
%GPS = Percent gross profit of sales
D = Dividend payments
%DR = Percent dividend rate
RE = Retained earnings at the end of the period
CAP = Capital assets needed during the period
CAS = Capital assets needed per sales dollar
FN = Funding needed during the period
TA = Total capital assets at the beginning of the period
E = Retained earnings at the beginning of the period

BD-03 DEBT FINANCING

$$\text{PMT factor} = (\text{Coupon})\ (\text{FACE})\ (1 - \text{TAX}) + [(\text{MP} - \text{FACE})/\text{N}]\ \text{TAX}$$

where

Coupon = Coupon interest rate per period
FACE = Face (par) value of the bond
TAX = Corporate tax rate (decimal)
MP = Market price of the bond
N = Number of periods until maturity

APPENDIX A

BD-04 INVESTMENT EVALUATION

Uneven Cash Flows

The internal rate of return is found using sophisticated numerical techniques, based on a method by E. Schroder. Reference: *Numerical Methods;* Dahlquist, Bjork, Anderson; pp 222-6, 237; Prentice-Hall, 1974.

$$PV = \sum_{j=1}^{N} CF_j \, (1 + i)^{-j}$$

$$FV = \sum_{j=1}^{N} CF_j \, (1 + i)^{N-j}$$

where

CF = Cash flow amount
i = Interest rate
N = Number of cash flows

Stock Flotation

The following is solved for i using an iterative process.

$$\text{Net income per share} = \sum_{k=1}^{N} \frac{Div_k}{(1+i)^k} + Div_{N+1} \sum_{k=N+1}^{\infty} \frac{(1+g)^{k-(N+1)}}{(1+i)^k}$$

where

i = Rate of return
k = Normal growth rate
g = Supernormal growth rate
Div = Dividend payment
N = Number of payments

BD-06 BREAKEVEN ANALYSIS

$$\text{Total Cost} = (\text{VAR CST})\ \text{UNIT}^{\left[1 + \frac{\log\left(1 + \frac{\text{LEARN}}{100}\right)}{\log 2}\right]} + \text{FIX CST}$$

$$\text{Total Revenue} = (\text{U PRICE})\ \text{UNIT}^{\left[1 + \frac{\log\left(1 + \frac{\text{EROSION}}{100}\right)}{\log 2}\right]}$$

$$GPM = \left[\frac{\text{Total Revenue} - \text{Total Cost}}{\text{Total Revenue}}\right] \times 100$$

APPENDIX A

The number of units required to achieve a specified GPM may be found by solving for the smallest number of units that satisfies the following.

$$(1 - \text{GPM}/100)\ (\text{Total Revenue}) - (\text{Total Cost}) = 0$$

The number of units required to achieve maximum GPM is found from

$$\text{UNIT} = \exp\left\{\frac{\left[\log \dfrac{c(g-1)}{a_1\ (f-g)}\right]}{1-f}\right\}$$

where

$$a_1 = \text{VAR CST}$$

$$c = \text{FIX CST}$$

$$g = \frac{-\log\left(1 + \dfrac{\text{EROSION}}{100}\right)}{\log 2}$$

$$f = \frac{-\log\left(1 + \dfrac{\text{LEARN}}{100}\right)}{\log 2}$$

BD-07 FACILITY CAPACITY

The following equations are used in computing the variables for option −1, series case:

$$\rho = \lambda/\mu$$

$$\text{Lq} = \frac{\rho^2}{(1-\rho)}k$$

$$\text{L} = \text{Lq} + k\rho$$

$$\text{Wq} = \text{Lq}/\lambda$$

$$\text{W} = \text{Wq} + k/\mu$$

$$\text{UTIL} = \rho$$

$$\text{IDLE} = (1-\rho)\text{W}$$

$$\text{TCW} = \text{CW} \times \text{Wq}$$

$$\text{TI} = \text{CI} \times \text{IDLE}$$

$$\text{TCI} = \text{TI}/\text{L}$$

$$\text{TC} = \text{TCW} + \text{TCI}$$

where

λ = Customer arrival rate

μ = Service rate

APPENDIX A

ρ = Utilization factor for servers
k = Number of phases or servers
Lq = Expected number of customers in queue
L = Expected number of customers in system
Wq = Expected waiting time per customer
W = Expected total time in system per customer
UTIL = Utilization factor for servers
IDLE = Expected facility idle time during a time interval equal to the expected time a customer is in the system
TCW = Total waiting time cost per customer
CW = Cost of customer waiting time per time period
TCI = Total facility idle time cost per length of time each customer is in the system
CI = Cost of facility idle time per time period
TC = Total cost per customer
TI = Total cost for facility idle time

The following equations were used in computing the variables for option 0, parallel case with exponential service rate:

$$\rho = \lambda / c\mu$$

$$a = \lambda / \mu$$

$$m = \frac{\left[\dfrac{ca^c}{c!\ (c - a)}\right]\left[\dfrac{1}{\mu\ (c - a)}\right]}{\left[\displaystyle\sum_{r=0}^{c-1} \frac{a^r}{r!} + \frac{ca^c}{c!\ (c - a)}\right]}$$

$$Wq = m$$

$$W = Wq + 1/\mu$$

$$Lq = m\lambda$$

$$L = Lq + \lambda/\mu$$

$$IDLE = (1 - \rho)W$$

$$TCW = CW \times Wq$$

$$TI = CI \times IDLE$$

$$TCI = TI/L$$

$$TC = TCW + TCI$$

$$UTIL = \rho$$

where

c = number of servers

APPENDIX A

The following equations were used in computing the variables for option 1, parallel case with constant service rate:

$$\rho = \lambda / c\mu$$

$$a = \lambda / \mu$$

$$m' = \left[\frac{c}{c+1}\right]\left[\frac{(1-\rho^{c+1})}{1-\rho^{c}}\right] \times m$$ where m is as shown in above case.

$$Wq = m'$$

$$W = Wq + 1/\mu$$

$$Lq = \lambda \times m'$$

$$L = Lq + \lambda/\mu$$

$$IDLE = (1-\rho)W$$

$$TCW = CW \times Wq$$

$$TI = CI \times IDLE$$

$$TCI = TI/L$$

$$TC = TCW + TCI$$

$$UTIL = \rho$$

BD-08 ECONOMIC REORDERING AND PRODUCTION RUNS

$$EOQ = \left\{\frac{2\ DPY\ [CPO + E(DDLT>R) \times St]}{CPU \times UHC}\right\}^{1/2}$$

$$EPQ = \frac{EOQ}{\sqrt{1 - \frac{DPY}{P}}}$$

$$Pr = \frac{CPU \times UHC \times (EOQ \text{ or } EPQ)}{DPY \times St}$$

$$St = \frac{CPU \times UHC \times (EOQ \text{ or } EPQ)}{DPY \times Pr}$$

where

EOQ = Economic Order Quantity

DPY = Demand quantity per year

CPO = Cost per order placed

E(DDLT>R) = Expected number of units greater than R required during lead time

St = Stockout cost per unit

CPU = Cost per unit

UHC = Unit holding cost as decimal fraction of CPU

EPQ = Economic production quantity

APPENDIX A

Pr = Probability of stockout

P = Production quantity per year

Demand during lead time is assumed to be normally distributed. The program finds a value z such that

$$\mathrm{Pr} = \int_{z}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx$$

Then R = EDDLT + $z\sigma$

and $$E(DDLT>R) = \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{R-EDDLT}{\sigma}\right)^2} + \frac{EDDLT-R}{\sqrt{2\pi}} \int_{\frac{R-EDDLT}{\sigma}}^{\infty} e^{-u^2/2}\, du$$

where

EDDLT = Expected demand during lead time

R = Reorder point that minimizes TAC

σ = Standard deviation of demand during lead time

For the order model the total annual cost (TAC) is:

$$TAC = \left[\left(\frac{EOQ}{2}\right) \times CPU \times UHC\right] + CPO\left(\frac{DPY}{EOQ}\right) + \left(St \times E(DDLT>R) \times \frac{DPY}{EOQ}\right)$$

$$+ [(R - EDDLT) \times CPU \times UHC] + (CPU \times DPY)$$

For the production model:

$$TAC = \left[\frac{EPQ}{2} \times CPU \times UHC \left(1 - \frac{DPY}{P}\right)\right] + CPO\left(\frac{DPY}{EPQ}\right)$$

$$+ \left[St \times E(DDLT>R) \times \frac{DPY}{EPQ}\right] + [(R - EDDLT) \times CPU \times UHC]$$

$$+ (CPU \times DPY)$$

BD-09 REORDER TIMING

$$\mathrm{Pr} = \frac{C_h \times T \times P}{Cb} \qquad Cb = \frac{C_h \times T \times P}{\mathrm{Pr}}$$

Demand during lead time is assumed to be normally distributed. The program finds a value of k such that:

$$\mathrm{Pr} = \int_{k}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx$$

APPENDIX A

$$\mu' = \mu + \mu(T/t)$$

$$\sigma' = \sqrt{\sigma^2 + (T/t)\sigma^2}$$

Find the value of k such that $\frac{1}{\sqrt{2\pi}} \int_k^\infty e^{-u^2/2}\, du = Pr$

$$R = \mu' + k\sigma'$$

$$ERC = C_r/T$$

$$EPC = C_p/T$$

$$EHC = C_h P\,[\,P - \mu - (DT/2)]$$

$$SS = R - \mu - DT$$

$$Eb = \frac{1}{T}\left[\sigma' \frac{1}{\sqrt{2\pi}}\, e^{-k^2/2} + Pr\,(\mu' - R)\right]$$

$$EBC = Eb \times Cb$$

$$TC = ERC + EPC + EHC + EBC$$

where

Pr = Desired probability of a stockout in any period

Cb = Cost of issuing a backorder

T = Review period in years

P = Unit price

C_h = Annual cost per unit of holding inventory, decimal fraction of unit price

μ = Expected demand during lead time

t = Lead time in years

σ = Standard deviation of demand during lead time

R = Optimum inventory position after placing an order

C_r = Cost to perform a review

ERC = Expected review cost per year

C_p = Cost to place an order

EPC = Expected annual cost for placing orders

EHC = Expected annual cost for holding inventory

D = Demand per year

SS = Safety stock

Eb = Expected number of backorders per year

EBC = Expected backorder cost per year

TC = Total annual cost of inventory system

APPENDIX A

BD-10 DEMAND FORECASTING

Level Coefficient Model

Initialize Function

$$F_t = \frac{\sum_{i=1}^{t} D_i}{t}$$

Revise Function

$$F_{t+1} = F_t + \alpha_F (D_t - F_t)$$

Forecast Function

$$DF_{t,t+k} = F_{t+1}$$

where

α_F = Level smoothing constant (0 to 1)
F_t = Level coefficient for period t
F_{t+1} = Level coefficient for period t + 1
D_t = Actual demand in period t
$DF_{t,t+k}$ = Forecasted demand at time period t for period t + k where k = 1, 2, 3,...

Level and Trend Coefficient Model

Initialize Function

$$T_t = \frac{D_t - D_1}{t - 1}$$

$$F_t = \frac{\sum_{i=1}^{t} D_i}{t} + \frac{(t-1)\, T_t}{2}$$

Revise Function

$$F_{t+1} = F_t + \alpha_F (D_t - F_t - T_t) + T_t$$

$$T_{t+1} = T_t + \alpha_T (F_{t+1} - F_t - T_t)$$

Forecast Function

$$DF_{t,t+k} = F_{t+1} + k\, T_{t+1} \qquad k = 1, 2, 3, \ldots$$

where

α_F = Level smoothing constant (0 to 1)
α_T = Trend smoothing constant (0 to 1)
T_t = Trend coefficient for period t
T_{t+1} = Trend coefficient for period t + 1
F_t = Level coefficient for period t
F_{t+1} = Level coefficient for period t + 1
D_t = Actual demand for period t
$DF_{t,\,t+k}$ = Forecasted demand at time period t for time period t + k where k = 1, 2, 3, ...

Level and Seasonal Coefficient Model

Initialize Function

$$F_t = \frac{\sum_{i=1}^{M} D_i}{M}$$

$$S_i = \frac{D_i}{F_t} \quad \text{for } i = 1, \ldots, M$$

Revise Function

$$F_{t+1} = F_t + \alpha_F \left(\frac{D_t}{S_{t,\,old}} - F_t \right)$$

$$S_{t,\,new} = S_{t,\,old} + \alpha_S \left(\frac{D_t}{F_{t+1}} - S_{t,\,old} \right)$$

Forecast Function

$$DF_{t,\,t+k} = F_{t+1}\,(S_k) \qquad k = 1, \ldots, M$$

where

α_F = Level smoothing constant (0 to 1)
α_S = Seasonal smoothing constant (0 to 1)
F_t = Level coefficient for period t
F_{t+1} = Level coefficient for period t + 1
D_i = Historical demand in period i
D_t = Actual demand in period t

APPENDIX A

$S_{t,\,old}$ = Old value for seasonal coefficient in period t

$S_{t,\,new}$ = New value for seasonal coefficient in period t

S_i = Initial value for seasonal coefficient for period i

S_k = Seasonal coefficient for period k

$DF_{t,\,t+k}$ = Forecasted demand at time period t for time period t + k where k = 1, 2, 3, ..., M

Level, Trend, and Seasonal Coefficient Model

Initialize Function*

$$T_t = \frac{\sum_{i=\tau+1}^{\tau+M} D_i - \sum_{i=1}^{M} D_i}{\tau M}$$

$$F_t = \frac{\sum_{i=\tau+1}^{\tau+M} D_i}{M} + \frac{(M-1)\,T_t}{2}$$

$$S_i = \frac{D_i}{F_t - (t-i)\,T_t} \qquad \text{for } i = 1, \ldots, t$$

Revise Function

$$F_{t+1} = F_t + \alpha_F \left(\frac{D_t}{S_{t,\,old}} - T_t - F_t \right) + T_t$$

$$T_{t+1} = T_t + \alpha_T \,(F_{t+1} - F_t - T_t)$$

$$S_{t,\,new} = S_{t,\,old} + \alpha_S \left(\frac{D_t}{F_{t+1}} - S_{t,\,old} \right)$$

Forecast Function

$$DF_{t,\,t+k} = (F_{t+1} + k\,T_{t+1})\,S_k \qquad k = 1, \ldots, M$$

where

Terms are defined as for previous model, plus

T_t = Trend coefficient for period t

T_{t+1} = Trend coefficient for period t + 1

APPENDIX A

*Note that $\tau + M$ causes a type of wrap-around sum. For example,

$\sum_{i=1}^{M} D_i$ uses the historical data for every period whereas

$\sum_{i=\tau+1}^{\tau+M} D_i$ uses the new actual demand data for those periods for which it has been input.

APPENDIX B

APPENDIX B: PROGRAM REFERENCE DATA

Program Number	Title	No. of Steps	Data Reg. Used	Flags Used	SBR Levels	Paren. Levels	Calls Pgm.	Special Functions Used	x ≷ t	ABS Address	Fix Decimal Format	Program Number
01	Module Check	45	1–6		0	0			CP	X	9	01
02	Long-Term Fin. Reg.	467	6–28	0	1	3			X	X	2	02
03	Debt Financing	60	1–5		0	2			X	X	9	03
04	Investment Eval.	373	0–9,11–16+	1	2	3			X	X	2,3,9	04
05	Proj. Plan/Budget	573	0–7, 8+	0–4,6	2	2			X	X	2	05
06	Breakeven	438	4–20		1	3			X	X	0,2,9	06
07	Facility Capacity	432	1–23	0	2	3			X	X	4,9	07
08	EOQ/EPQ	944	0–30+	0,1	3	3			X	X	0,2,4,9	08
09	Reorder Timing	352	0–3,5–28	1,2	4	2	08		X	X	0,2,4,5,9	09
10	Demand Forecasting	703	0–10,29–59	0,1	1	3			X	X	0,3,5,9	10
11	Line Balancing*	579	0–9, 10+	0-2	0	2			X	X	1,9	11
	Pointers and Counters	34										

*Does not run in ENG mode.

ONE YEAR LIMITED WARRANTY FOR CALCULATOR AND/OR LIBRARY MODULE

WARRANTEE: This Texas Instruments Electronic Calculator Warranty extends only to the original purchaser of the calculator or module.

WARRANTY DURATION: This Texas Instruments Electronic Calculator and/or Library Module is warranted to the original purchaser for a period of one (1) year from the original purchase date.

WARRANTY COVERAGE: This Texas Instruments Electronic Calculator and/or Library Module is warranted against defective materials and workmanship. **THIS WARRANTY IS VOID IF: (I) THE CALCULATOR OR MODULE HAS BEEN DAMAGED BY ACCIDENT OR UNREASONABLE USE, NEGLECT, IMPROPER SERVICE OR OTHER CAUSES NOT ARISING OUT OF DEFECTS IN MATERIAL OR WORKMANSHIP, (II) THE SERIAL NUMBER HAS BEEN ALTERED OR DEFACED.**

WARRANTY PERFORMANCE: During the above one (1) year warranty period your defective calculator or module will either be repaired or replaced with a reconditioned model of an equivalent quality (at TI's option) when the calculator or module is returned, postage prepaid and insured, to a Texas Instruments Service Facility listed below. In the event of replacement with a reconditioned model, the replacement unit will continue the warranty of the original unit or 90 days, whichever is longer. Other than the postage and insurance requirement, no charge will be made for such repair, adjustment, and/or replacement.

WARRANTY DISCLAIMERS: ANY IMPLIED WARRANTIES ARISING OUT OF THIS SALE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO THE ABOVE ONE (1) YEAR. TEXAS INSTRUMENTS SHALL NOT BE LIABLE FOR LOSS OF USE OF THE CALCULATOR OR OTHER INCIDENTAL OR CONSEQUENTIAL COSTS, EXPENSES OR DAMAGES INCURRED BY THE PURCHASER.

Some states do not allow the exclusion or limitation of implied warranties or consequential damages, so the above limitations or exclusions may not apply to you.

LEGAL REMEDIES: This warranty gives you specific legal rights, and you may also have other rights that vary from state to state.

TEXAS INSTRUMENTS CONSUMER SERVICE FACILITIES

Texas Instruments Service Facility
P.O. Box 2500
Lubbock, Texas 79408

Texas Instruments Service Facility
41 Shelley Road
Richmond Hill, Ontario, Canada

Consumers in California and Oregon may contact the following Texas Instruments offices for additional assistance or information:

Texas Instruments Consumer Service
3186 Airway Drive Bldg. K
Costa Mesa, California 92626
(714) 540-7190

Texas Instruments Consumer Service
10700 Southwest Beaverton Highway
Park Plaza West, Suite 11
Beaverton, Oregon 97005 (503) 643-6758

IMPORTANT NOTICE REGARDING PROGRAMS AND BOOK MATERIALS

TEXAS INSTRUMENTS MAKES NO WARRANTY, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR PARTICULAR PURPOSE, REGARDING THESE PROGRAMS OR BOOK MATERIALS OR ANY PROGRAMS DERIVED THEREFROM AND MAKES SUCH MATERIALS AVAILABLE SOLELY ON AN "AS-IS" BASIS.

IN NO EVENT SHALL TEXAS INSTRUMENTS BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE PURCHASE OR USE OF THESE BOOK MATERIALS OR PROGRAMS AND THE SOLE AND EXCLUSIVE LIABILITY TO TEXAS INSTRUMENTS, REGARDLESS OF THE FORM OF ACTION, SHALL NOT EXCEED THE PURCHASE PRICE OF THIS BOOK. MOREOVER, TEXAS INSTRUMENTS SHALL NOT BE LIABLE FOR ANY CLAIM OF ANY KIND WHATSOEVER AGAINST THE USER OF THESE PROGRAMS OR BOOK MATERIALS BY ANY OTHER PARTY.

TEXAS INSTRUMENTS
INCORPORATED
DALLAS TEXAS

Printed in U.S.A

1014984-9