HEWLETT PACKARD

HP 82482A

Finance Pac

Owner's Manual

For the HP-71

Notice

Finance Pac

Owner's Manual

For Use With the HP-71

December 1983

82482-90001

Printed in Singapore

Introducing the Finance Pac

The HP-71 Finance Pac programs have been designed to easily perform the functions of a powerful financial calculator while offering the computer advantages of continuous memory, file creation and storage, and the potential use of peripherals.

One program, the time value of money program (TVM), solves a wide range of problems for any person who works with financial contracts of any type: borrowers, lenders, lessors, lessees, accountants, real estate brokers, automobile dealers, and investors, to name a few. Here are some of the things you can do with the program:

- Solve for the number of periods, the interest rate, the present value, the payment amount, or the future value of a uniform series of payments.
- Calculate an amortization schedule of a loan.
- Enter an uneven series of cash flows from the keyboard or from data files in memory. The cash flows may be grouped or ungrouped.
- Review or edit an uneven cash flow series.
- Calculate the internal rate of return (*IRR*) or the net present value (*NPV*) of an uneven cash flow series.
- Store an uneven cash flow series as a data file in memory.

The other Finance Pac program, the depreciation program (DEP), enables you to calculate, display, print, and store depreciation schedules in memory. You can choose from five different methods to calculate depreciation.

In "How to Use This Manual," on page 7, you will find information on how best to approach your study of this powerful financial tool.

The programs in this pac are based on programs written for Hewlett-Packard by G. Marc Choate and Michael L. Hand.

Contents

How To Use This Manual

This manual is both a learning and a reference tool. It is designed to enable you to rapidly learn to use all the features of the Finance Pac and to serve as a reference with examples to the different financial functions. The information in this manual assumes that you have read sections 1 and 6 of the *HP-71 Owner's Manual.*

For an introduction to the Finance Pac and how it works, read through section 1, "Getting Started," of this manual. Then, depending on which financial functions you want to do, read and work through the examples in sections 2 through 4, which describe the time value of money program, or section 5, "Depreciation."

At the end of the manual there are five appendixes for your information. Appendix A, "Owner's Information," includes warranty and service information. Appendix B, "Error Messages," describes error messages. Appendix C, "Creating Data Files," describes how to write your own BASIC language programs to create or read data files that can be used by the TVM program. Appendix D, "File Names," gives the names of the files used in the Finance Pac. Appendix E, "Financial Formulas," lists the financial formulas. These are followed by a comprehensive subject index.

An overlay has been included in the package to help you use the time value of money program. The mnemonics on the overlay will remind you of the keyboard location of the TVM functions.

Section 1

Getting Started

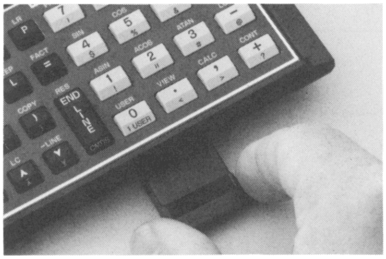
Installing and Removing the Finance Pac Module

The finance module can be plugged into any one of the four ROM ports on the front edge of the computer.

CAUTIONS

- Be sure to turn off the HP-71 (press [f][OFF]) before installing or removing the module.
- If you have removed a module to make a port available for the finance module, before installing the finance module, turn the computer on and then off to reset internal pointers.
- Do not place fingers, tools, or other foreign objects into any of the ports. Such actions could result in minor electrical shock hazard and interference with pacemaker devices worn by some persons. Damage to port contacts and internal circuitry could also result.

To insert the module, orient it so that the label is right side up, hold the computer with the keyboard facing up, and push the module into the slot until it snaps into place. During this operation be sure to observe the precautions described above.



To remove the module, use your fingernail to grasp the lip on the bottom of the front edge of the module and pull it straight out of the port. Install a blank module in the port to protect the contacts inside.

Running the Programs in the Finance Pac

Before continuing on, install the finance module in one of the ports, turn the computer on, and place the finance overlay on the keyboard. The overlay shows the positions on the keyboard of several financial functions. Above each key is the function performed when that key is pressed. Functions in blue require you to press the blue shift key ([g]) before pressing the specified key. To reinforce the finance concepts, the keystroke procedures in this manual will use the financial functions. In other words, we will use [n] to refer to the Q key, [i%] to refer to the W key, etc. An illustration of the overlay is on page 14. You will want to keep the overlay with your finance module to help you easily reference the finance functions.

There are two programs in the Finance Pac, TVM (Time Value of Money) and DEP (DEPreciation). To run either one, type the word RUN followed by the program name, and press [END LINE]. To show how simple and convenient it is to use the programs, we invite you to begin running the TVM program using the following examples.

Example 1: Suppose that you plan to make regular deposits at the end of each year to a retirement account that will accumulate at annual compound interest. You are interested in having $250,000 in the account at the end of 20 years. What annual payment must be made over the next 20 years if the account earns interest at a 9% annual rate?

Without trying to memorize the keystroke sequence, please follow the procedures below.

Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :■`	Displays the program introduction or help prompt. Whenever you see this prompt, certain keys are redefined to perform financial functions. Refer to page 13 for a description of the keys that are active while the prompt is showing.
20 [n]	Stores the number of deposits that you will make into the retirement account. When you pressed [n], the letter Q did not appear in the display. The program executed a routine that stored 20 as the number of periods (deposits).
`n = 20.00 :■`	Shows that 20 is stored as the number of deposits (periods).

Input/Result

9 [i%]	Stores the annual interest rate as a percentage.
`i% = 9.00 :█`	Shows that 9 is stored as the periodic interest rate.
250000 [FV]	Stores the amount that you wish to accumulate, the future value.
`FV = 250000.00 :█`	Shows that $250,000 is stored as the future value.
[PMT]	Calculates and displays the amount of the annual payment. The minus sign indicates that you must pay this money into the account in order to receive the accumulated amount in 20 years.
`PMT = -4886.62 :█`	Let the program continue to run to solve the next example.

Example 2: If the account earns interest at a 9.6% annual rate, what annual payment is necessary to accumulate the desired amount in your retirement fund? Since only the interest rate has changed, you do not need to reenter the previous financial data.

9.6 [i%]	Stores the new interest rate as a percentage.
`i% = 9.60 :█`	Shows that 9.6% is stored as the new interest rate.
[PMT]	Calculates and displays the required annual payment.
`PMT = -4567.28 :█`	
[Exit]	Exits the program.
`TVM ended`	

In these examples you calculated an annual payment that would be required to accumulate a future value of $250,000 under two different circumstances. At 9% annual interest the required annual payment is $319.34 more than at 9.6% interest.

The full explanation of the TVM program begins in section 2 and continues through section 4. The explanation of DEF begins in section 5. After studying the rest of the material in this section, you can proceed to any of these sections. However, sections 3 and 4 assume that you are familiar with section 2.

Setting the Display Rate

You should use the DELAY command (refer to "Controlling Program Display Speed" in section 1 of the *HP-71 Owner's Manual*) to set the display rate. Error messages and some prompts are affected by this rate. The Finance Pac programs do not change the display DELAY rate.

Experiment with several DELAY settings. If the rate is too slow, you may find that pressing [END LINE] to override the DELAY is inconvenient. If the DELAY rate is too short, you may miss a displayed message. Set a DELAY rate that is comfortable for you. A DELAY rate of 8 seconds or greater is infinity to the machine. In this case it is necessary for you to press [END LINE] to permit the program to proceed with further calculations.

Using the [RUN] Key

To begin running a Finance Pac program you type RUN, followed by either TVM or DEF, then press [END LINE]. Thereafter, when you press the [RUN] key, the HP-71 will run the most recently run program. This means that if you have run and exited TVM, you can begin running it again by pressing [RUN]. However, to change from one program to the next, you must type RUN, followed by the program name, then press [END LINE].

System Requirements

The programs in the Finance Pac do not require any of the peripherals that are available for the HP-71. Each result is available to you in the display, and the file handling capabilities of the HP-71 can be used to store cash flow information.

The programs have routines that can print information in formatted tables if a printer is attached to the HP-71 using the HP 82400A HP-IL Interface Module (refer to section 13 of the *HP-71 Owner's Manual*). If you have a video interface for the HP-71 (and the HP-IL interface), the Finance Pac prompts and calculated results will be intelligible, but the programs do not have any formatting specifically designed to be used with a video display.

Section 2

TVM: The Five Financial Variables

Introduction

If you are given a choice between receiving $1 today and receiving $1 tomorrow, your decision is very easy to make and anyone can predict what your choice will be.

If the choice is between receiving $1 today and receiving more than $1 tomorrow, the decision is not at all easy, nor can it be predicted very well. If the choice is between $1 today and $1.10 tomorrow, some persons might choose the $1.10 tomorrow, others the $1 today. If the period of time becomes a month or a year the decision becomes very difficult.

Time value of money (TVM) is a conceptual model that economists and mathematicians have developed to help the person who must make decisions where there are choices like "$1 now or $A later." The model has gained acceptance as a valid decision-making aid in a wide range of financial situations such as: evaluating a consumer loan contract, calculating payments to a retirement account, or evaluating a real estate investment.

The TVM program in the Finance Pac presents a versatile set of routines that will solve complex financial calculations. The routines permit you to use the power of the computer to store financial information, recall and use that information later, and use information that you generate in other programs, including those that you write yourself.

This section describes one aspect of the TVM program—the five financial variables and the active keyboard.

The TVM Keyboard

If the finance module is in the HP-71, and you type RUN TVM and press [END LINE], you will see the following display:

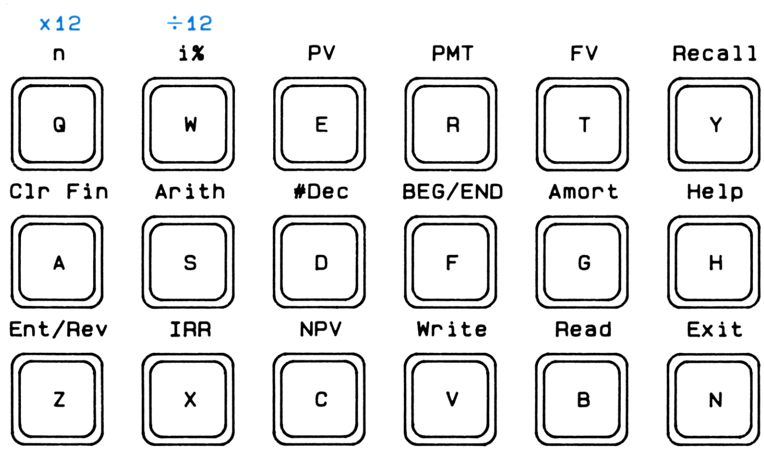
```
For help, press H :█
```

This is the program introduction (or help) prompt. When you see this prompt, certain keys have been redefined to the TVM functions (refer to the following facsimile of the keyboard). This special state of the keyboard will be called *active*.

When you press [Help] (the [H] key) the program enters a routine that displays lines of information about the TVM program. After you press [Help], use the up- and down-arrow keys ([▲] and [▼]) to move forward and backward through the messages, one line at a time. If you press a key that executes a specific routine (such as the [IRR] key), the display will show the first line of a description of the function of that key. You can then use the arrow keys to display the entire message, one line at a time. If you press [Exit] or [ON], the program returns to the program help prompt.

You used the active keyboard if you followed the procedure to solve the retirement account example on page 10. When you entered 20 and pressed [n], neither the lower- nor the uppercase letter Q appeared in the display. Rather, the program ran a routine that stored 20 as the number of periods. When you pressed [PMT], the letter R did not appear. Rather, a portion of the TVM program calculated the payment amount and the answer appeared in the display. At the end of the procedure you pressed [Exit]. Again, no corresponding letter N appeared. Rather, you left the program.

The following is a facsimile of the upper-left portion of the HP-71 keyboard with the overlay in place. Above each key is noted the function performed when that key is pressed while the keyboard is active. You will recognize a few of them as keys you used to solve the example in section 1.



The TVM Keyboard

In addition to the keys above, [EEX] puts the letter E into the display for entering numbers in exponential (scientific) notation.

You can press [Exit] (the [N] key) to end the program. The display will show the following:

`TVM ended`

When you exit the TVM program, the display setting is either two decimal places or the number of decimals specified with [#Dec] (refer to page 21). In addition, DEFAULT ON is set (refer to section 2 in the *HP-71 Owner's Manual*).

If you wish to temporarily suspend the program, with the intention of returning to it soon, press [ON]. The display will show:

`TVM suspended`

All financial information that you have entered is still in the machine and is accessible to the program. You can resume by pressing [f][CONT].

If the keyboard is active and you do not press a key for some length of time, the machine does not turn off, but enters a low power state. If you exit the program by pressing [Exit] or [ON], the machine will turn itself off after 10 minutes of inactivity.

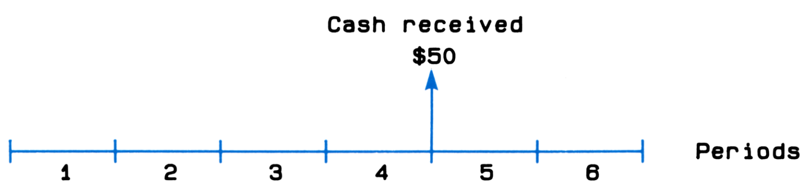
> **Note:** If there are characters in the display to the right of the colon when you press [ON], the display will clear and leave the colon and cursor at the left of the display. The TVM program is still running. If you press the [ON] key when the display to the right of the colon is clear, the program stops running but can be restarted by pressing [f][CONT]. If the program is restarted, you will see the help prompt in the display and the data in the program will be unchanged.

Cash Flow Diagrams

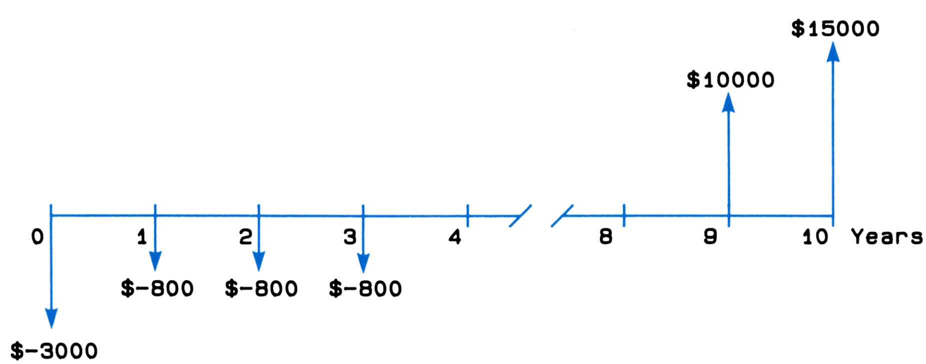
The concepts and examples presented in this section are representative of a wide range of financial calculations. If your specific problem is not illustrated in the following pages, *don't* assume that the options in TVM are not available to solve it. Every financial calculation involves certain basic elements. However, the terminology used to refer to these elements typically differs among the various segments of the business and financial communities. To use the TVM program, all you need to do is identify the basic elements in your problem, then structure the problem so that it will be apparent which quantities you need. The *cash flow diagram* is an invaluable aid in doing this.

The diagram is simply a pictorial representation of the timing and direction of cash flows in a transaction. It begins with a horizontal line, called a *time line*. The time line represents the duration of a financial transaction and is divided into intervals that represent *time periods*. The time period used is the period over which interest is compounded: months for monthly compounding, years for annual, and so forth. For example, the time line for a financial problem that covers six months looks like this:

Cash transactions are represented by vertical arrows placed along the time line. Cash you receive is represented by an arrow pointing up (positive) from the time line. Cash you pay out is represented by an arrow pointing down (negative) from the line. The payments are placed at a position on the line that represents the points in *time* where the payments occur. For example, the time line below can be interpreted as $50 received at either the *end* of period 4 or the *beginning* of period 5:



Example: The following cash flow diagram represents a financial transaction in which the investor pays or receives cash at various times. He makes an initial investment of $3,000, makes payments of $800 at the end of years 1, 2, and 3, neither pays nor receives anything in years 4 through 8, receives a payment of $10,000 at the end of year 9, and receives a final payment of $15,000 at the end of year 10.



The Cash Flow Sign Convention

Whenever you use the TVM program you must use the cash flow sign convention. There are two helpful hints for remembering how the sign convention works:

- Look at the problem from the viewpoint of either the borrower or the lender/investor. Do not use both viewpoints in the same problem.
- Cash received is represented by a positive number and an arrow that points up. Cash paid out is represented by a negative number and an arrow that points down.

The *borrower* initially receives cash (positive numbers and arrows that point up from the time line) and pays cash later (negative numbers and arrows that point down from the time line).

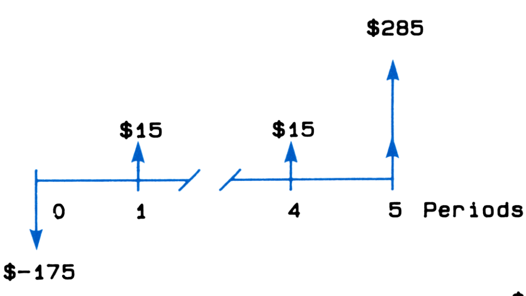
The *investor/lender* initially pays cash (negative numbers and arrows that point down from the time line) and receives cash later (positive numbers and arrows that point up from the time line).

Uniform Series of Payments

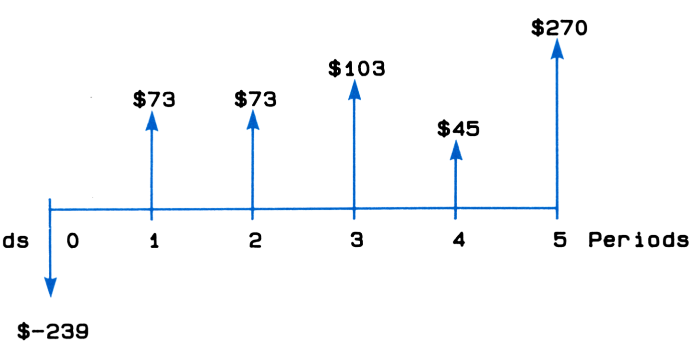
The type of time value of money problem that is solved most frequently is the *uniform series of payments* problem, which consists of an uninterrupted series of equal payments over more than one period. There is often an additional payment on either end or both ends of the series. The TVM program solves a wide variety of uniform series problems.

The TVM program also solves another general type of cash flow problem. It is called the *uneven series of payments* problem. The discussion of that type of problem begins on page 53.

The cash flow diagrams below illustrate the difference between uniform and uneven series problems.



Uniform Series



Uneven Series

The Financial Variables

Each problem involving a uniform series of payments has five variables associated with it. When you know any four of the variables, you can use the TVM program to calculate the fifth (unknown) variable. To learn about these five variables, consider the following example.

Example: Some years ago you borrowed $45,000 on a mortgage at an annual interest rate of 8.25%. The monthly payment is $−388.07. If you have made 151 payments, what is the balance that you owe on the loan? (The keystroke solution begins on page 19, but we will explain the variables here.)

The *number of periods* is called *n*. These periods are of equal length—all months, all quarters, all years, and so forth. In the example, $n = 151$ months. You use the [n] key to store and to calculate *n*. You can enter the exact number of periods, or you can enter the number of years followed by the * (multiply) symbol and the number of compounding periods per year. Or, you can use [g][x12] to store 12 times the number of years (the number of months). *n* must be greater than 0.

The *periodic interest rate percent* is called *i%*. If *n* is in years, the rate is an annual compound interest rate. If *n* is in months, the rate is a monthly compound interest rate. Interest rates are usually quoted as *annual interest rates* (also called *nominal annual interest rate* or *interest rate per year*), and must be divided by the number of compounding periods per year to find the periodic compounded interest rate. In the example above, the 8.25% annual rate is 8.25%/12 compounded monthly. You use the [i%] key to store and to calculate *i%*. The interest rate must be for the time period that you used when storing or calculating *n*. You can enter the periodic *i%* exactly, or you can enter the annual *i%* followed by the / (divide) symbol and the number of compounding periods per year. If the period length is one month, you can also use [g][÷12], which automatically divides the annual *i%* by 12 and stores it. *i%* must be greater than 0.

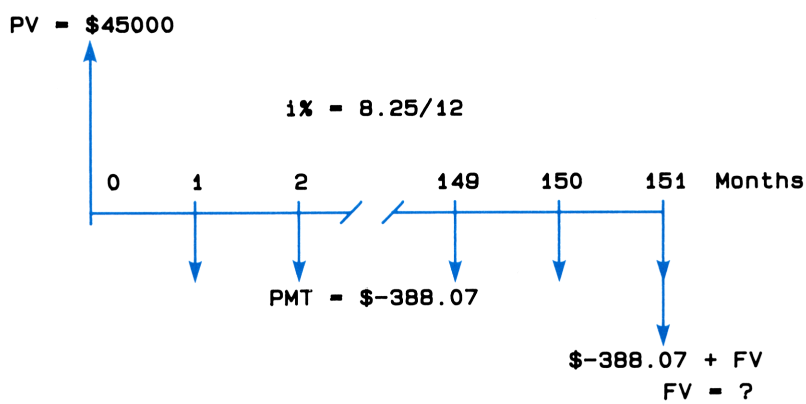
> **Note:** Many problems have monthly payments but the interest rate and number of periods are given in years. To help you in the conversion use [g][x12] to store 12 times the number of years entered, and [g][÷12] to store the monthly interest rate when you enter the annual interest rate.

The cash flow at time zero on the time line is called *PV*, which stands for *present value*. To the lender it is the loan amount. To the borrower it is the amount borrowed. To the investor it is the initial investment. The *PV* is diagrammed at the start of the time line. The *PV* can be zero, positive, or negative. In the example above, $PV = \$45,000$. You use the [PV] key to calculate and to store *PV*. If payments are being made at the beginning of each period (refer to the discussion below), *PV* is in addition to the first *PMT*.

The uniform series of payments is called *PMT*. The payments are the same amount each period and there are no skips or changes. The payment amount can be zero, positive, or negative. The payments can occur at either the END of each period or the BEGinning of each period (refer to page 40). In the example, $PMT = \$-388.07$ and the payments are made at the end of each period. You use the [PMT] key to calculate and to store *PMT*. The [BEG/END] key switches between BEGinning and END payment modes (refer to page 21). The TVM program begins running with the assumption that the payment mode is END.

The cash flow at the end of the time line in a uniform series of payments is called *FV*, which stands for *future value*. In a lease it may be the residual value of the equipment. In a loan it is the balloon payment or remaining balance of the loan. The *FV* always occurs at the end of period *n*, the last period in the problem. The *FV* can be zero, positive, or negative. In the example, *FV* is the unknown variable. You use the [FV] key to calculate and to store *FV*. If payments are at the end of each period, *FV* is in addition to the last *PMT*.

The cash flow diagram for the example, drawn from the viewpoint of the borrower, is the following:



Using the TVM program, the solution to the example is:

Input/Result

RUN TVM [END LINE] — Begins running the program.

```
For help, press H :▒
```

Displays the help prompt.

151 [n] — Stores 151 as *n*.

```
n =   151.00 :▒
```

Input/Result

Input/Result	
8.25 [g] [÷12]	Stores 8.25 divided by 12 as the periodic *i%*.
`i% = 0.69 :▒`	
45000 [PV]	Stores 45000 as *PV*.
`PV = 45000.00 :▒`	
-388.07 [PMT]	Stores −388.07 as *PMT*.
`PMT = -388.07 :▒`	
[FV]	Calculates and displays the remaining balance of the loan. The balance of the loan is $24,237.07.
`FV = -24237.07 :▒`	
[Exit]	Exits the TVM program.*
`TVM ended`	

* Many of the procedures in this manual begin with RUN TVM and end by exiting the program. This allows you to begin running the examples at any point, but does add extra keystrokes. In a situation where you are going from one problem to the next, you may wish to clear the variables (using [Clr Fin]) before beginning a new computation, rather than exiting the program.

The top and middle rows of keys contain eight functions that relate to the five financial variables as follows:

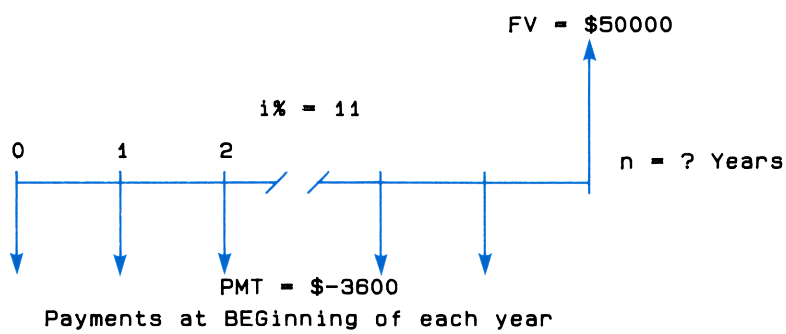
Key	Description
[Clr Fin]	Sets *i*%, *PV*, *PMT*, and *FV* to their initial state of 0 and *n* to its initial state of 1. When you press [Clr Fin], the prompt `Clr n,i%,PV,PMT,FV` is displayed and the keyboard is active.
[Arith]	Performs arithmetic on the displayed financial variable. If, for example, `i% = 0.70` is the monthly interest rate in the display and you want to know what the yearly percentage is, type `*12` [Arith] and the product, `i% *12 = 8.40 :▓` will be displayed. At this point, the program is running, the stored variable has *not* been altered, and the keyboard is active. To use the [Arith] key, you input the operator first, then the number, and press [Arith].
[#Dec]	Changes the display setting for rounding numbers. When you type `RUN TVM` [END LINE], the program sets the display to round numbers to two decimal places. If you want a different setting, press [#Dec]. The display will show `Enter # of decimals`, and you respond with a number and press [END LINE]. The number of decimal places entered is displayed. The display setting will remain until you change it again or until you rerun the `TVM` program. The range of values for this display setting is the same as the `FIX` function. (Refer to section 2 of the *HP-71 Owner's Manual*.)
[BEG/END]	Switches between BEGinning and END of period payment mode. Refer to page 40 for information on how to use this key.
[Amort]	Calculates the amount of interest and change in balance of a loan. This is the amortization function and it is explained starting on page 43.
[Help]	Displays brief instructions on using the `TVM` program.
[Recall]	Recalls the value of one of the five financial variables. When you press [Recall], the word `RECALL` is put into the display. Then press the key for any of the five variables and the current value will be displayed. (Although the current value is displayed, it cannot be used in display calculations.) You can also recall the payment mode setting by pressing [Recall] [BEG/END].
[EEX]	Puts the letter `E` into the display when entering numbers in exponential (scientific) notation. For example, seven million can be entered as `7000000` or `7E6`, which means seven times ten raised to the sixth power. (Refer to "Exponential Notation" in section 2 of the *HP-71 Owner's Manual*.)

Calculating the Number of Periods

To solve for *n* (the number of compounding periods in a uniform series problem) you first need to store the periodic *i%*, *PV*, *PMT*, and *FV*. Then press [n] to calculate the number of periods (*n*).

Example: If you deposit $3,600 in a bank at the beginning of each year, how long will it take to accumulate $50,000, assuming an annual interest rate of 11%?

The cash flow diagram is as follows:



Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :▒`	
11 [i%]	Stores 11% as the interest rate.
`i% = 11.00 :▒`	
0 [PV]	Stores 0 as *PV*.
`PV = 0.00 :▒`	
-3600 [PMT]	Stores −3600 as *PMT*. Remember the minus sign.
`PMT = -3600.00 :▒`	

Input/Result

50000 [FV] — Stores 50000 as *FV*.

```
FV =   50000.00 :▩
```

[BEG/END] — Switches between END of period payment mode and BEGining of period payment mode. The program is in END mode when it is first run. Pressing [BEG/END] changes the mode back and forth: END to BEGin, BEGin to END with each press of the [BEG/END] key.

```
Now in BEGin mode :▩
```

[n] — Calculates and displays the number of periods. Let the program continue to run.

```
n =   8.29 :▩
```

The 8.29 answer means that at least 8 yearly payments of $3,600 must be made to accumulate to $50,000. The 0.29 fraction *does not* mean that a ninth payment of 0.29 × $3,600 is needed. The mathematics for the uniform series type problem does not calculate the partial payment. You can easily continue the example to calculate the accumulated amount if there are 8 or 9 *full* payments. Continue directly from the previous keystroke procedure.

Input/Result

```
n =   8.29 :▩
```

Answer still displayed from above. (Assumes that the TVM program is still running and the keyboard is active.)

8 [n] — Stores 8 as the number of periods.

```
n =   8.00 :▩
```

[FV] — Calculates *FV*.

```
FV =   47390.30 :▩
```

The answer is the accumulation of 8 full payments of $3,600.

Input/Result

9 [n] — Stores 9 as the number of periods.

```
n =   9.00 :▓
```

[FV] — Calculates *FV*. The answer is the accumulation of 9 full payments of $3,600.

```
FV =   56599.23 :▓
```

[Exit] — Exits the program.

```
TVM ended
```

Calculating the Periodic Interest Rate

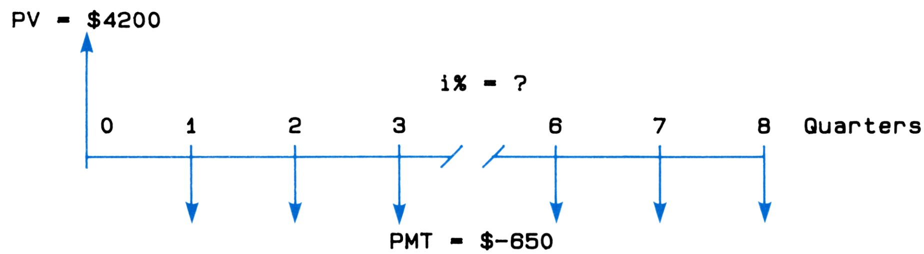
You calculate the periodic interest rate of a uniform series of cash flows by first storing *n*, *PV*, *PMT*, and *FV*, then pressing the [i%] key.

The program searches for positive solutions to *i%*. If the cash flow problem is viewed from the investor's viewpoint, there is a unique positive *i%* solution when there is one change of sign in the series of cash flows and the sum of the cash flows is greater than zero.* Such a series will be called a *conventional series* of cash flows.

* For other cases, called *unconventional series*, there may be no solution. Or there may be more than one solution, in which case the interpretation of the results is the subject of much discussion in finance. The routines in TVM permit you to solve the more complex situations (for an example refer to page 77), but this section limits the discussion to the simpler cases.

Example 1: You are offered two alternatives by the owner of a used truck. You can pay either $5,100 cash, or $900 now plus 8 quarterly payments of $650. If the quarterly payment offer is treated as a loan, what is the annual interest rate? (Hint: With the $900 down, the actual amount of the loan is $4,200 and there are 8 loan payments of $650. The result of the calculation will be a quarterly interest rate that must be multiplied by 4 to obtain a nominal annual interest rate.)

The following cash flow diagram is presented from the viewpoint of the borrower.



Input/Result

RUN TVM [END LINE] Begins running the program.

```
For help, press H :
```

8 [n] Stores 8 as *n*.

```
n =   8.00 :
```

4200 [PV] Stores 4200 as *PV*.

```
PV =   4200.00 :
```

-650 [PMT] Stores −650 as *PMT*.

```
PMT = -650.00 :
```

Input/Result

[i%]	Begins running the interest routine.
`Calculating` `i% = 5.01 :█`	The calculation takes several seconds. The result is the quarterly interest rate.
*4 [Arith]	Multiplies the quarterly interest rate by 4 and displays the annual interest rate.
`i% *4 = 20.03 :█`	
[Exit]	Exits the program.
`TVM ended`	

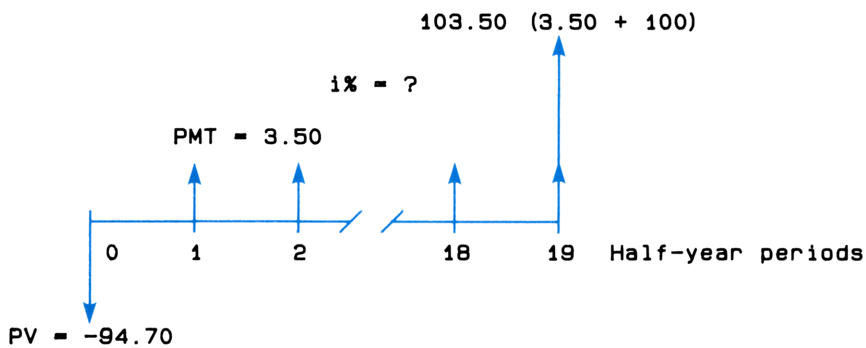
The [Arith] key allows you to perform calculations using the variables stored by TVM without interrupting the program. For more information on [Arith] refer to page 21.

Example 2: What is the yield on a semi-annual coupon bond if the settlement date is on a coupon date,* maturity is in 9.5 years, the coupon is 7%, and the bond is purchased at 94.70?

The bond contract promises a uniform series of cash flows to the bond holder. To solve the problem we need to extract the relevant uniform cash flow variables from the conventional bond market wording. The bond price is quoted as if the face of the bond were \$100. We can use the 100 figure as the amount that will be paid at maturity; thus, $FV = 100$. Each half year the bond will pay a coupon payment of one-half of the stated annual coupon rate. The semi-annual payment will be one-half of 7 percent of 100 or 3.50. Thus $PMT = 3.5$. These are the cash flows that the bond holder will receive. He will receive the 3.50 payment 19 times (9.5 years times 2 payments each year) plus the 100 maturity amount at the end of the nineteenth period. Thus, $n = 19$. For this privilege he will pay the bond price of 94.70; thus $PV = -94.7$.

* The TVM program solves for the price or yield of a bond if the settlement date falls on a coupon date. The TVM routines assume that the period lengths are equal.

The following cash flow diagram is presented from the viewpoint of the investor. The solution will be a semi-annual interest rate, which must be multiplied by two to give the nominal annual bond yield.



Input/Result

RUN TVM [END LINE] — Begins running the program.

```
For help, press H :■
```

19 [n] — Stores 19 as *n*.

```
n =   19.00 :■
```

-94.7 [PV] — Stores −94.7 as *PV*. Don't forget the minus sign for the investment.

```
PV = -94.70 :■
```

3.5 [PMT] — Stores 3.5 as *PMT*.

```
PMT =   3.50 :■
```

Input/Result

100 [FV] — Stores 100 as *FV*.

```
FV =   100.00 :■
```

[i%] — Begins running *i%* solution.

```
Calculating
Answer: i% =   3.90 :■
```

Calculation takes several seconds. The semi-annual yield is calculated and displayed.

2 [Arith] — Multiplies the semi-annual yield by 2 to obtain the nominal annual bond yield.

```
i% *2 =   7.80 :■
```

[Exit] — Exits the program.

```
TVM ended
```

Calculating the Present Value

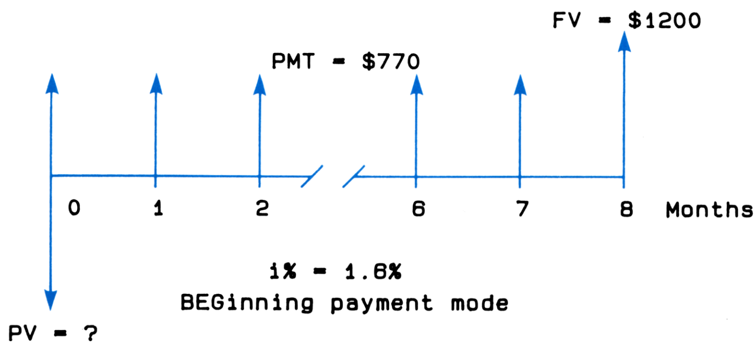
To calculate the present value of a uniform series of cash flows, you need to know the uniform series payment amount (*PMT*), the cash flow at the end of the series (*FV*), the number of regular payments (n), and the periodic interest rate (*i%*). If you know these four values, you can enter them into the TVM program in any order and press [PV] to solve for *PV*.

Because of the HP-71's continuous memory, you must pay particular attention to *FV*. In *PV* (as well as *PMT*) solution problems, *FV* is often zero, and you must make sure that *FV* in the program is, in fact, zero. When TVM is first run, *FV* is set to zero. If you continue from one problem to the next without exiting the program, you may need to use the [Clr Fin] key to clear the financial variables to their default values (all zeroes, except for n, which is set to 1). You could also set a financial value to zero by entering 0 and pressing a financial key. In the examples in this manual, *FV* is stored as zero except when RUN TVM [END LINE] is used.

* The semi-annual yield must be multiplied by two to obtain the nominal annual bond yield, the conventional method of yield quotation in the bond market.

Example 1: In the settlement of an account, your customer offers to pay 8 payments of $770 at the beginning of each month plus a balance of $1,200 at the end of the eighth month. If your alternative investments must yield a 1.6% monthly interest rate, what is the present value of the series of payments being offered to you? (Hint: Remember that these payments are made at the beginning of the period. You must use the [BEG/END] key to switch from END of period payment mode to BEGinning of period payment mode.)

The following cash flow diagram is presented from the viewpoint of the investor. At the end of the procedure let the program continue to run if you wish to continue with example 2.



Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :■`	
8 [n]	Stores 8 as *n*.
`n = 8.00 :■`	
1.6 [i%]	Stores 1.6 as the monthly rate.
`i% = 1.60 :■`	
770 [PMT]	Stores 770 as *PMT*.
`PMT = 770.00 :■`	

Input/Result

1200 [FV]	Stores 1200 as *FV*.
`FV =   1200.00 :■`	
[BEG/END]	Switches back and forth from END to BEGinning of period payment mode.
`Now in BEGin mode :■`	
[PV]	Calculates and displays the present value of the series of cash flows.
`PV = -6887.85 :■`	Let the program continue to run in order to do the next example.

The offer is equivalent to an alternative investment worth $6,887.85 assuming an investment rate of 1.6% per month.

Example 2: If your counter-offer to example 1 is that the final payment increase to $2,000, what is the present value of the series of payments?

The procedure below assumes that the program is still running from the previous calculation.

Input/Result

`PV = -6887.85 :■`	Displays the answer from the previous calculation.
2000 [FV]	Stores new value of *FV*.
`FV =   2000.00 :■`	
[PV]	Calculates and displays *PV* based on the new *FV*. The numbers for the other variables are unchanged from the previous example.
`PV = -7592.45 :■`	

Input/Result

[Exit]	Exits the program.

```
TVM ended
```

The revised present value is $7,592.45. Increasing the amount of the balloon payment by $800 increases the value of the investment by $704.60.

The second example demonstrates that you can change any variable that you used previously (as long as you are still running the TVM program). The five financial variables can be changed and stored in any order, and any of the five can be calculated.

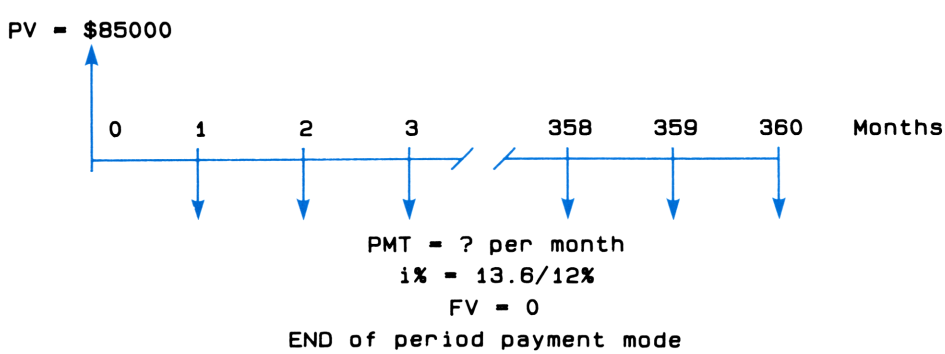
Be sure to use the cash flow sign convention when storing *PV*, *PMT*, and *FV*. Otherwise, the cash flow series that is stored may result in an error when you solve for *i%*. Refer to the discussion on page 17. If such an error occurs the program permits you to review numbers and enter the correct number without interrupting the program. You can review any financial variable simply by pressing [Recall] and then the financial key you want to review. To correct the number, type in the new value and press the financial key to store the value.

Calculating the Payment Amount

Perhaps the most frequently solved problem of the uniform series type is that of calculating the amount of a payment that is equivalent to a known present value or future value. Loan and lease payments, as well as many other applications, fall into this category.

Example 1: What is the monthly loan payment for an $85,000 conventional mortgage fully amortized over 30 years if the annual percentage rate is 13.6%? (Hint: We will use the [g] [×12] and [g] [÷12] key options of TVM to store the number of months and the monthly rate.)

The cash flow diagram is presented from the viewpoint of the borrower.



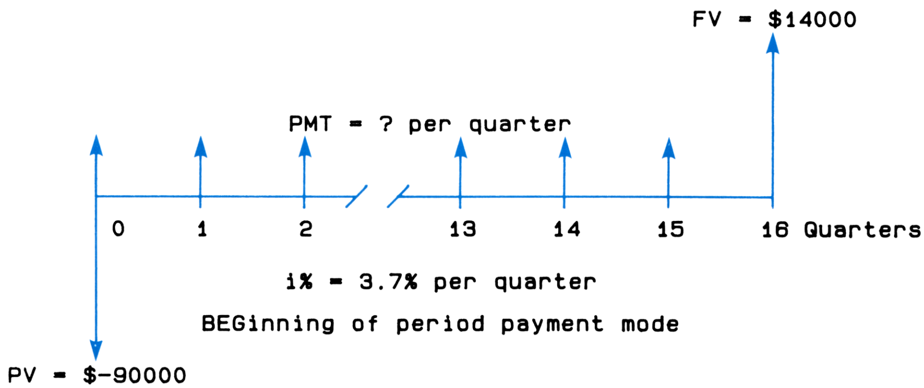
Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :█`	
30 [g][×12]	Enters the number of years. The program multiplies the number of years by 12 and stores the product (the number of months) as *n*.
`n = 360.00 :█`	
13.6 [g][÷12]	Enters the annual interest rate. The program divides the annual rate by 12 and stores the resulting monthly rate.
`i% = 1.13 :█`	
85000 [PV]	Stores *PV*.
`PV = 85000.00 :█`	
[PMT]	Calculates and displays the payment.
`PMT = -980.29 :█`	
[Exit]	Exits the program.
`TVM ended`	

The monthly payment on an $85,000, 30 year mortgage at 13.6% annual interest rate is $980.29.

Example 2: A piece of equipment has a value of $90,000. The equipment will be leased for four years. Lease payments will be made at the beginning of each quarter. The estimated salvage value of the equipment at the end of the lease is $14,000. There are no other cash flows to be considered. Estimate the amount of the lease payment if the lessee's borrowing rate of 3.7% per quarter is used as the interest rate of the lease.

The cash flow diagram is presented from the viewpoint of the lessor.



Input/Result

RUN TVM [END LINE] Begins running the program.

```
For help, press H :■
```

[BEG/END] Switches to BEGinning of period payment mode.

```
Now in BEGin mode :■
```

16 [n] Stores n.

```
n =   16.00 :■
```

3.7 [i%] Stores $i\%$.

```
i% =   3.70 :■
```

-90000 [PV] Stores PV. Don't forget the minus sign.

```
PV = -90000.00 :■
```

Input/Result

14000 [FV]	Stores *FV*.
`FV = 14000.00 :■`	
[PMT]	Calculates and displays the amount of the lease payment.
`PMT = 6650.73 :■`	
[Exit]	Exits the program.
`TVM ended`	

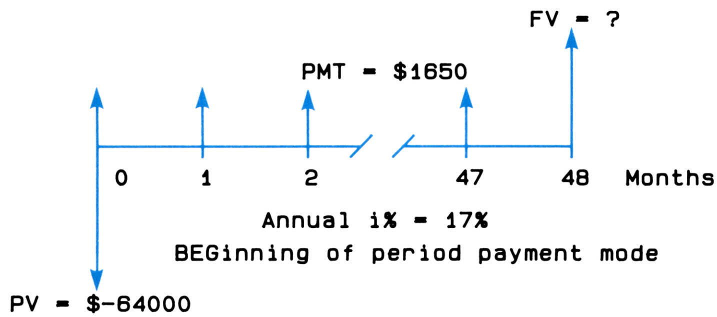
Calculating the Future Value

You are able to solve many useful uniform series problems where the future value (*FV*) of the series is unknown. Many are extensions of the problems in which *PV* or *PMT* are solved, such as calculating the balloon payment (remaining balance of a loan), estimating the residual value of leased property, or calculating the accumulation of a retirement account balance.

As in the other problems involving the five financial variables, the solution requires that you store four of the variables and solve for the fifth. In this case the fifth is *FV*, which is solved by pressing [FV].

Example 1: You wish to lease a piece of equipment to a client willing to pay $1,650 at the beginning of each month for 48 months. Your investment is $64,000 and your nominal annual borrowing rate is 17%. What is the amount of the residual necessary to achieve at least a 17% yield?

The cash flow diagram is presented from the viewpoint of the lessor. We will use the [g][×12] and [g][÷12] keys to enter years and annual interest.



Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :▮`	
[BEG/END]	Switches to BEGinning of period payment mode.
`Now in BEGin mode :▮`	
4 [g][×12]	Enters the number of years and multiplies by 12 to store 48 as *n*.
`n = 48.00 :▮`	
17 [g][÷12]	Enters the annual rate and divides by 12 to store the monthly *i*%.
`i% = 1.42 :▮`	
-64000 [PV]	Stores *PV*. Remember to enter the minus sign.
`PV = -64000.00 :▮`	

Input/Result

1650 [PMT] Stores *PMT*.

```
PMT =   1650.00 :■
```

[FV] Calculates and displays *FV*.

```
FV =   11801.69 :■
```

[Exit] Exits the program.

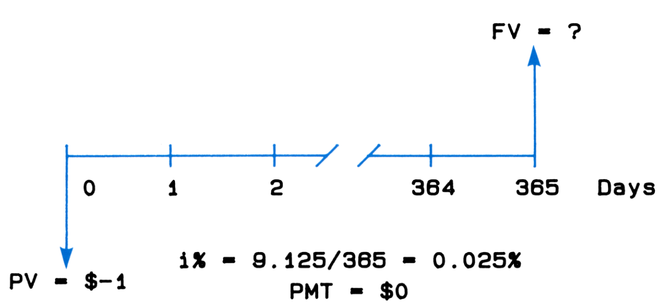
```
TVM ended
```

The residual value would have to exceed $11,801.69 for the capitalized value of the lease payments and residual value to be greater than the investment.

We will now look at an example that may not immediately appear to be a uniform series cash flow problem. We will calculate an equivalent effective annual interest rate when interest is compounded daily.

Example 2: A financial institution quotes the earnings rate on the investment program to be "nominal annual interest rate, compounded daily." Because of daily compounding, the effective annual rate is higher than the quoted nominal annual rate. If the quoted nominal annual rate is 9.125%, what is the effective annual rate?

The cash flow diagram is drawn from the viewpoint of the investor. Note that the payment amount is 0. The problem is viewed as an investment of 1, with 365 payments of 0 at 9.125%/365 daily interest rate.



Note: This example includes instructions to set the display to show four decimal places.

Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :▒`	
[#Dec]	Prompts for decimal setting.
`Enter # of decimals ▒`	
4 [END LINE]	Sets decimal places to four.
`4.0000 :▒`	Displays four decimal places. TVM is now ready to accept input of the variables.
365 [n]	Stores *n*.
`n = 365.0000 :▒`	
9.125/365 [i%]	Stores *i*%.
`i% = 0.0250 :▒`	
-1 [PV]	Stores *PV*. Remember to enter the minus sign.
`PV = -1.0000`	
[FV]	Calculates and displays the amount that $1 would grow to in 365 days if interest is compounded daily.
`FV = 1.0955 :▒`	
[Exit]	Exits the program.
`TVM ended`	

The answer is the amount to which $1 would accumulate at daily compounded interest in 1 year, that is, $1.0955. The 9.55 cents is interest on the $1. This means that 9.125% nominal annual interest, compounded daily, is equivalent to 9.55% effective annual interest.

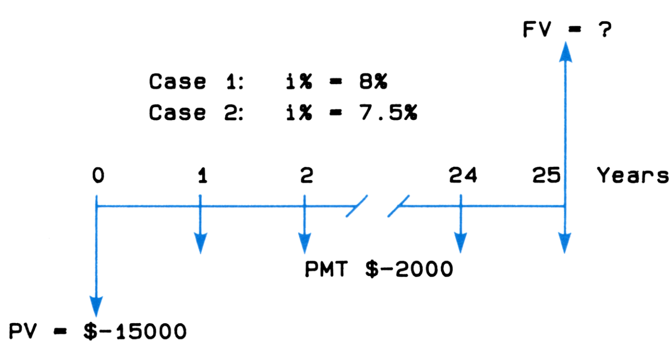
Technical Note: This calculation is the mathematical "daily compounding" equivalence method. The formula to get the answer above from a nominal annual interest rate percent is $(1 + i\%/36500)^{365}$. If you subtract 1 and multiply the difference by 100 you get the percentage rate. (An interest rate of 9.125% converts to 9.5530363%.)

You may see a reference to the term "continuous compounding." That is not the same as daily compounding. The formula to calculate the "continuous compounding" annual rate equivalent to a nominal annual percent rate is $e^{i\%/100}$ where e is the base of the natural logrithms (refer to "Logarithm Functions" in section 2 of the *HP-71 Owner's Manual*). Subtract 1 and multiply by 100 to get the percentage. (Using this method, 9.125% converts to 9.5542857%.)

"Continuous compounding" is sometimes used in reference to the formula $(1 + i\%/36000)^{365}$. You can solve the formula using the procedure in example 2 (page 36). At the start simply enter 9.125/360, rather than 9.125/365, before you press [i%]. The rest of the procedure is the same. (An interest rate of 9.125% converts to 9.6919325%.)

Example 3: Your investment account presently contains $15,000 and you contribute an additional $2,000 at the end of each year. In order to learn how sensitive your personal retirement plans are to financial market interest rates, calculate the amount your account will accumulate to in 25 years if the account earns an annual rate of 8% per year. What will the amount be if the rate is 7.5%?

The cash flow diagram is presented from the viewpoint of the investor. You may wish to let the program continue to run at the end of the procedure in order to work the next example.



Input/Result

RUN TVM [END LINE] — Begins running the program.

```
For help, press H :■
```

25 [n] — Stores *n*.

```
n =   25.00 :■
```

8 [i%] — Stores *i%*.

```
i% =   8.00 :■
```

-15000 [PV] — Stores *PV*.

```
PV = -15000.00 :■
```

-2000 [PMT] — Stores *PMT*.

```
PMT = -2000.00 :■
```

[FV] — Calculates and displays *FV* for the 8% case.

```
FV =   248939.01 :■
```

7.5 [i%] — Stores new value of *i%*.

```
i% =   7.50 :■
```

[FV] — Calculates and displays *FV* for the 7.5% case. Let the program continue to run if you wish to continue to the next example.

```
FV =   227430.82 :■
```

The two answers show that a drop in the interest rate of one-half of one percent will lower the accumulated investment amount by close to $21,500.

BEGinning/END of Period Payment Mode

You must pay attention to the point in time at which payments occur when you solve time value of money problems. The uniform series of cash flow problems solved by the TVM program assume that payments occur *either* at the BEGinning of each period or at the END of each period. In all examples in this manual, cash payments are assumed to occur at the END of each period, unless specifically stated otherwise.

Whether the payments occur at the beginning or at the end of the period *is* important in financial contracts. If you lease equipment to others, you are certainly disappointed if you learn that the lease agreement you just signed specifies that the lease payments are to be made at the *end* of each calendar quarter, rather than at the *beginning*, as you had planned!

The TVM program permits you to solve either problem quite easily, and to switch from one to the other without disturbing the rest of the program. You switch from one payment *mode* to the other by pressing [BEG/END]. The following two messages will be displayed alternately as you press [BEG/END]:

```
Now in BEGin mode :█
Now in END mode :█
```

The BEGinning/END mode key affects only the uniform series of payments *(PMT)* and does *not* affect *PV* or *FV*. The *PV* always occurs at the beginning of the first period, which is time zero, and *FV* always occurs at the end of time period *n*.

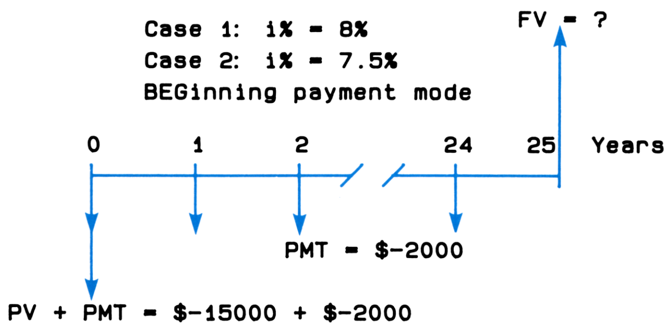
You can recall the payment mode status without changing it by pressing [Recall] [BEG/END]. Also, if the program is in BEGinning of period payment mode, a small digit 1, the "Flag 1 annunciator," will show in the display to the left of the PRGM annunciator. When the program is in END mode, the annunciator will not be on (the digit 1 does not appear).

You can see the effect of the payment mode by continuing from the previous example 3. In that example, the payment mode is END.

Example: Example 3 (page 38) is solved assuming that payments are made at the end of each period. What is the effect, for both 8% and 7.5%, if the payments are made at the beginning of each year?

If you have not already done so, solve example 3 on page 38 and continue directly to the procedure on the next page, without interrupting the program.

The following diagram presents the cash flows from the viewpoint of the investor with payments at the beginning of each year.



Input/Result

`FV = 227430.82 :▮`	Displays answer from previous example.
[BEG/END]	Switches to BEGinning of period payment mode.
`Now in BEGin mode :▮`	
8 [i%]	Enters 8% as *i%*.
`i% = 8.00 :▮`	
[FV]	Calculates and displays *FV* in BEGin mode at 8%.
`FV = 260635.96 :▮`	
7.5 [i%]	Stores new *i%*.
`i% = 7.50 :▮`	

Input/Result

FV	Calculates and displays *FV* in BEGin mode at 7.5%.

```
FV =  237627.50 :█
```

Exit

```
TVM ended
```

Moving all payments to the beginning of each period increases the total accumulation by nearly $11,700 at 8% and by nearly $10,200 at 7.5%. You are quite justified in being concerned about whether contracts call for payments at the beginning or at the end of periods. The TVM program confirms your concern.

Section 3

TVM: Amortization

Introduction

This section describes a second aspect of TVM, the process of amortizing a loan. Before continuing on, make sure that you are familiar with the concepts presented in sections 1 and 2.

Using [Amort]

When a loan is amortized, the payment amount is first used to pay the interest that has accrued.* The remainder of the payment is used to reduce the debt (the principal). In financial literature, this process is called the *actuarial* method, and the loan is called a *direct reduction* loan.

For example, the first $100 monthly payment on a $500 loan with monthly interest of 1.5% consists of: $7.50 interest and a $92.50 reduction in the loan amount. This leaves a remaining balance on the loan of $407.50.

To calculate amortization you should first store or calculate three numbers:

- The periodic interest rate ([i%]).
- The loan amount ([PV]).
- The periodic payment amount ([PMT]).

The BEGinning/END of period payment mode is selected and the cash flow sign convention is followed. The *n* and *FV* financial variables are not used in the amortization calculation. The program assumes that all periods are of equal length. (Partial period calculations are not permitted.)

* Technical literature refers to two methods of handling the accrual of interest: the actuarial method and the U. S. method. The actuarial method is used in all of the routines of the TVM program. In the actuarial method if the payment amount is not sufficient to cover the interest amount, the difference is added to the loan balance. The U. S. method does not add the difference to the loan balance.

When you press [Amort], the program begins a sequence of prompts that display the current values in the program for *i%*, *PV*, *PMT*, and BEGinning/END payment mode. The current values will be displayed with all of the decimal places available for the variable. If you wish to enter the entire number just as it appears, press [END LINE]. If you wish to change the number, press [ON] to clear the number in the display, then reenter the entire number. Or, use the arrow keys to move to the digits you wish to change, type over them, and press [END LINE].

The program then displays the prompt `# of pmts: 1`. The number you enter will be the total number of payments to be included in the amortization schedule (the default value is `1`). If you want only the first year of a monthly-payment loan, you enter `12`. If you want all of the payments for a 30 year monthly-payment loan, you enter `360`.

The program then displays the prompt `Increment 1`. The number you enter will tell the program how many payments to amortize before the results are displayed. If you want to see the amortization amounts for each payment, you enter `1` (or just press [END LINE] and let it default to `1`). If you want to see only an annual summary of a monthly-payment loan, you enter `12`.

If an HP 82401A HP-IL Interface and a printer are connected and operating with the HP-71, you will see a prompt asking whether you wish the results to be displayed (`D`) or printed (`P`). If there is no printer, this prompt is skipped.

The program then displays:

- The period number.
- The amount to interest.
- The change in the loan balance.
- The balance of the loan at the end of that period.

The output begins with period 0 (the status at the beginning of the loan), and proceeds through the total number of periods that you entered. You see each successive item of output by pressing [END LINE].

You can exit the sequence at any time by pressing [Exit] or [ON]. Pressing either of these keys interrupts the program to display `Done :■`, leaving the keyboard active.

Technical Note: The program includes two methods of calculating amortization. The method that is automatically used is the "dollars and cents" method, where all calculations are in dollars and cents. However, if you wish to have the program use all of the decimal places (internal calculations are not rounded), use the following procedure:

1. After entering *i%*, *PV*, and *PMT*, but before pressing [Amort], press [ON] to suspend the program.
2. Type `SFLAG 10` [END LINE] and press [f][CONT].

This procedure sets flag 10 and continues the program. Then press [Amort]. The program tests the status of flag 10. If it is set, the program uses all decimal places available in the amortization calculation. The display will still show the number of decimal places set with the [#Dec] key, but more places will be carried internally. A printed schedule will print nine decimal places.

Displayed Results

Example 1: For income tax purposes you wish to see the annual summary of amount to interest and change in balance of a loan. The loan is for $5,000, has monthly payments of $−135.00, and carries an annual interest rate of 14.4%. You wish to see the annual summary for a four-year period.

Hint: Use the top row of keys ([i%], [PV], and [PMT]) to store the financial variables. Use the cash flow sign convention. In the sequence of prompts at the start of the routine, enter 48 as the number of payments and 12 as the increment since the example asks for an annual summary for a four-year period.

Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :■`	Displays the help prompt. Now store the financial data.
14.4 [g][÷12]	Stores the monthly interest rate. Divides the annual rate by 12 and stores the monthly rate as *i%*.)
`i% = 1.20 :■`	
5000 [PV]	Stores the loan amount as *PV*.
`PV = 5000.00 :■`	
-135 [PMT]	Stores the payment amount as *PMT*. (Remember the minus sign.)
`PMT = -135.00 :■`	

Input/Result

[Amort]	Begins running the amortization routine. You will see a sequence of prompts that can be edited or left to their current values (or default values). After a prompt appears, you can edit the value. Then press [END LINE] once to enter the value and go to the next prompt.
`Int. rate % 1.20000000`	
[END LINE]	Enters the interest rate and displays the loan amount (*PV*).
`Loan amt. 5000.0000000`	
[END LINE]	Enters the loan amount and displays the payment amount (*PMT*).
`Pmt. amt. -135.0000000`	
[END LINE]	Enters the payment amount and displays the current payment mode.
`Pmt. mode (B/E) E`	
[END LINE]	Enters the payment mode and displays the prompt for the total number of payments to be covered in the amortization calculations (default value is 1).
`# of pmts. 1`	
48 [END LINE]	Enters 48 as the total number of payments to be amortized and displays the prompt for the number of payments between each calculation.
`Increment 1`	

Input/Result

12 END LINE

Enters 12 as the increment for an annual summary and begins the output sequence. Once the output sequence begins, END LINE is used to view the display output, starting with the 0 period index.*

```
Period 0
```

END LINE

Displays the amount of interest at time 0.

```
Amt. of Int. -0.00
```

END LINE

Displays the change in the loan balance. (If payments were made at the beginning of the period, this value would be the payment amount.)

```
Chg. in Bal.  0.00
```

END LINE

Displays the loan balance at time 0, the beginning of the loan.

```
End. Bal.  5000.00
```

END LINE

Displays the period number. Since the previous item was at time 0, this line will show a summary of 12 months.

```
Period 12
```

END LINE

Displays the amount of interest over the 12 month period.

```
Amt. of Int. -658.16
```

* If you have a printer connected, you will see an additional prompt asking whether you wish the results to be displayed or printed.

Input/Result

END LINE	Displays the change in the loan balance over the 12 month period.
`Chg. in Bal. -961.84`	
END LINE	Displays the loan balance at the end of period 12.
`End. Bal. 4038.16`	
END LINE	Proceeds to next item.
`Period 24`	
END LINE	Continue pressing END LINE to display the remaining summary.
⋮	
END LINE	Displays the loan balance at the end of 48 months.
`End. Bal. 169.85`	
END LINE	Displays prompt for continuing calculations. (The previous ending balance becomes the beginning balance at time 0 for any further calculations.)
`More? (Y/N) ■`	
N END LINE	A no (N) answer displays the end of the routine, with TVM running and the keyboard active.
`Done :■`	
Exit	Exits the program.
`TVM ended`	

Printed Schedule

Example 2: Calculate a quarterly amortization summary for the first two years (24 months) of a $9,500 loan with monthly payments (made at the end of each month) of $−245. The annual interest rate is 11.76%.

The following procedure assumes that the HP 82401A HP-IL Interface is attached to the HP-71 and that there is an 80-column printer operating in the HP-IL loop. If you do not have a printer, you will see the results in the display of your HP-71, but in the format of the previous example. If you have a 24-column printer, the printed table will have the same information as the table in this example, but it will be formatted for the narrower paper.

Input/Result

RUN TVM [END LINE] — Begins running the program.

```
For help, press H :■
```

Displays the help prompt. Now store the financial data.

11.76 [g][÷12] — Stores the monthly interest rate. (Divides the annual rate by 12 and stores the monthly rate as *i%*.)

```
i% =  0.98 :■
```

9500 [PV] — Stores the loan amount as *PV*.

```
PV =  9500.00 :■
```

−245 [PMT] — Stores the payment amount as *PMT*. (Remember the sign convention.)

```
PMT = -245.00 :■
```

Input/Result

[Amort]	Begins running the amortization routine. You will see a sequence of prompts that can be edited or left to their present values (or default values). After a prompt appears, you can edit the value. Then press [END LINE] once to enter the value and go to the next prompt.
`Int. rate % 0.98000000`	
[END LINE]	Enters the interest rate and displays the loan amount (*PV*).
`Loan amt. 9500.0000000`	
[END LINE]	Enters the loan amount and displays the payment amount (*PMT*).
`Pmt. amt. -245.0000000`	
[END LINE]	Enters the payment amount and displays the current payment mode.
`Pmt. mode (B/E) E`	
[END LINE]	Enters the payment mode and displays the prompt for the total number of payments to be covered in the amortization calculations (the default value is 1).
`# of pmts. 1`	
24 [END LINE]	Enters 24 as the total number of payments to be amortized and displays the prompt for the number of payments between each calculation.
`Increment 1`	

Input/Result

3 [END LINE] — Enters 3 as the increment for a quarterly summary and displays the prompt for selecting the method of presenting the results.

```
Disp or Prt? (D/P) ■
```

(This prompt will not be displayed unless a printer is operating in the HP-IL loop with the HP-71.)

P [END LINE] — Selects printed output. At this point the table is printed. When it is finished, the next prompt asks if you want to proceed with more calculations on the same loan.

```
More? (Y/N) ■
```

N [END LINE] — Selects the option to exit the routine. The TVM program is still running and the keyboard is active.

```
Done :■
```

[Exit] — Exits the program.

```
TVM ended
```

The printed quarterly summary appears as follows:

```
Loan balance is             9500
Payment amount is           -245
     made at the End of each period
Periodic interest rate % is 0.98000000000
```

Period #	Amt. of Int.	Change in Balance	Ending Balance
0	-0.00	0.00	9500.00
3	-274.82	-460.18	9039.82
6	-261.16	-473.84	8565.98
9	-247.09	-487.91	8078.07
12	-232.61	-502.39	7575.68
15	-217.69	-517.31	7058.37
18	-202.33	-532.67	6525.70
21	-186.52	-548.48	5977.22
24	-170.23	-564.77	5412.45

If you select the Y option when the amortization routine displays More? (Y/N) ■, the routine returns to the sequence of prompts for the financial information. The current values are displayed again in sequence for you to edit or leave unchanged. Notice, however, that the amount of the loan will now be the loan balance at the end of the amortization routine you just completed. This permits you to run another segment of the amortization schedule.

The financial variables that are stored or calculated by the top row of keys (*i%*, *PV*, *PMT*) are not changed from their original values. This means that you can rerun the entire amortization procedure by merely exiting the routine and pressing [Amort] again.

Using these features, you can experiment and run a variety of amortization routines with the same basic information.

Section 4

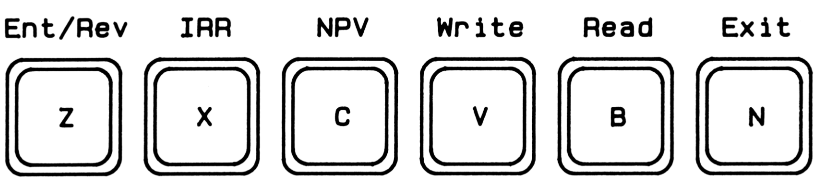
TVM: *NPV* and *IRR*

Introduction

Sections 1, 2, and 3 describe the many variations of a uniform cash flow series. This section describes another aspect of TVM: calculations involving an uneven cash flow series. Before continuing on, be sure that you are familiar with the concepts presented in sections 1 and 2.

Uneven Series of Cash Flows

The third row of keys is used to work with uneven cash flow series. Below is a facsimile of the left half of that row of keys with the financial overlay in place. The label above each key indicates the function performed by that key when the TVM program is running and the keyboard is active.



The table on the next page lists these keys and gives a brief description of the function of each.

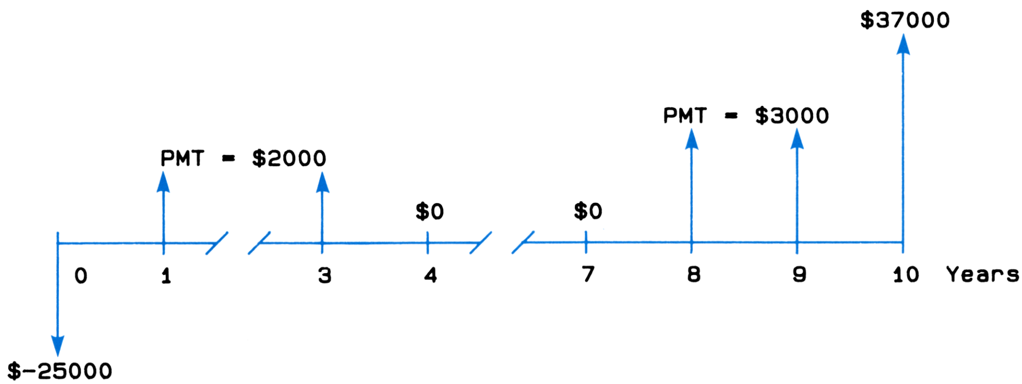
Key	Description
[Ent/Rev]	Used to enter an uneven series of cash flows from the keyboard. The series then becomes the *current* series of cash flows. [Ent/Rev] also permits you to review and edit the current series. Always use the cash flow sign convention when entering cash flows.
[IRR]	Calculates the internal rate of return *(IRR)* of the current uneven series of cash flows. Before you use this key you should use [Ent/Rev] to enter data or [Read] to read data. The explanation of *IRR* begins on page 70.
[NPV]	Calculates the net present value *(NPV)*, the net equivalent uniform series *(NEUS)* to the *NPV*, and the net future value *(NFV)* of the current uneven cash flow series. Before you use this key you should first enter data using [Ent/Rev] or read data using [Read]. The explanation of *NPV*, *NEUS*, and *NFV* begins on page 64.
[Write]	Writes the current cash flow series to a data file in the HP-71 memory, but does not change the current series available to keys [IRR] and [NPV].
[Read]	Reads a cash flow data file from the HP-71 memory into the TVM program and makes the data in that file the current cash flow series. This series can then be used to calculate *IRR* or *NPV* or can be reviewed and edited by using [Ent/Rev].
[Exit]	Exits the TVM program. The explanation of [Exit] begins on page 15.

An uneven series of cash flows is one with a changing pattern of cash payments over the term of the investment. The cash flow sign convention is used and payments are assumed to be made at the end of the period. The BEGinning/END mode key does not affect the calculations with uneven series payments.

You should use the cash flow diagram to organize the data for an uneven cash flow series. Refer to page 15 for information on diagramming cash flows. A short example is presented next.

Example: Draw the cash flow diagram for the following series: an investment of $25,000 followed by three annual cash inflows of $2,000, four years in which there are no inflows or outflows, two cash inflows of $3,000, and a final cash inflow of $37,000.

From the viewpoint of the investor, such an uneven series of cash flows is diagrammed as follows:



Uneven Cash Flow Table

Another way of presenting an uneven cash flow series is with a table such as the following. The table permits you to "diagram" a very long series of cash flows in less space than the time line cash flow diagram. We will call the table a *cash flow table.* The cash flows in the following table are taken from the example above.

j	*t*	*C(j)*	*N(j)*
0	0	−25000	1
1	1-3	2000	3
2	4-7	0	4
3	8-9	3000	2
4	10	37000	1

In the uneven cash flow table the first column is the group number, *j*. In the example there are four groups of cash flows *in addition to the initial group*. The initial group, group 0 ($j = 0$), is usually the investment amount (remember to use the cash flow sign convention). Every uneven cash flow series will have a group 0; it may be for $0, but it must be included.

The second column (t) is the time period or periods in which the cash flows for that group occur. For group 0, the cash flow occurs only once at time 0 (the end of period 0). For group 1 ($j = 1$) the cash flows occur in periods 1, 2, and 3. In uneven cash flow analysis, *cash flows always occur at the end of the period*. For example, if your problem has a cash flow occurring at the beginning of period 11, place the cash flow on the cash flow diagram at the *end* of period 10. The BEGinning/END of the period payment mode is not involved in uneven cash flow analysis.

The third column is the amount of the cash flow in group j and is headed $C(j)$. The cash flow sign convention is used.

The fourth column is the number of consecutive times that a cash flow in group j occurs, and is headed $N(j)$. Typically there is only 1 cash flow in the initial group, which is to say that $N(0) = 1$. (However, there may be more—and this is permitted within the program.) In group 1 the $2,000 cash flow occurs 3 times, as indicated in the table and on the diagram. Thus $N(1) = 3$.

All values of $N(j)$ should be integers (no fractional periods) that are greater than zero. Noninteger values for any $N(j)$ will result in calculated numbers that are not meaningful in finance.

The number of cash flows that can be used in the program is limited only by the size of memory available in RAM. If there is not sufficient memory to manipulate the cash flows, the program displays:

```
Not enough memory
```

You can then interrupt the program by pressing the [ON] key, CATalog the HP-71 RAM files (refer to section 6 of the *HP-71 Owner's Manual*), PURGE files that are not needed, press [f][CONT] to return to the TVM program, and try the procedure again.

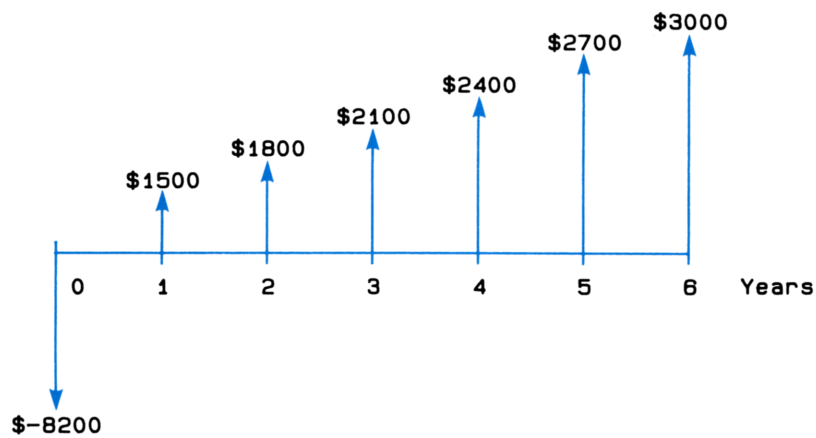
Grouped and Ungrouped Uneven Cash Flow Series

There are two types of uneven cash flow series: grouped and ungrouped. If the uneven series has cash flows that repeat from one period to the next consecutive period, as in the case of three $2,000 payments in the example above, the uneven series is called a *grouped* uneven series of cash flows. If the uneven series has no repeating consecutive cash flows the series is called an *ungrouped* series. When you enter uneven cash flows into the TVM program using [Ent/Rev], a prompt will be displayed that permits you to select either grouped or ungrouped cash flow entry. If your cash flow series is ungrouped, you will use fewer keystrokes by using the ungrouped option.

The following information will be used in each of the next few examples to demonstrate the uneven cash flow capabilities of the TVM program.

Example: George offers you an investment opportunity for an investment of $8,200. The investment will return six annual payments at the end of each year, starting with $1,500 and increasing by $300 each year. Diagram the cash flows and list them in a cash flow table.

The series is an ungrouped uneven series of cash flows. No two consecutive cash flows are equal to one another. The diagram and table are as follows:



Group *j*	Time Period *t*	Cash Flow *C(j)*	# of Periods *N(j)*
0	0	−8200	1
1	1	1500	1
2	2	1800	1
3	3	2100	1
4	4	2400	1
5	5	2700	1
6	6	3000	1

Notice that the *N*(*j*) column of the table is merely a series of 1's. That is the distinguishing characteristic of ungrouped series. In running the TVM program for a grouped series of cash flows, you must enter the *N*(*j*) value for each group. However, if you elect the *ungrouped* option, the TVM program automatically enters the series of 1's. You can use either option to enter this series of cash flows into the TVM program, but you will likely find the ungrouped option more convenient. For uneven series that include a relatively large number of periods and several groups, you may find the grouped option more convenient.

You will also notice the similarity between a uniform series of cash flows and an uneven series. In the example on the previous page, if the cash flows for time periods 1 through 6 are the same amount, the ungrouped uneven series is identical to a uniform series.

Entering Cash Flows From the Keyboard

To enter a cash flow series from the keyboard use [Ent/Rev] and follow the prompts. Unless your cash flow series is a simple one, you will find that writing the cash flows in the cash flow table format is helpful. Write them in table format, then enter the numbers from the two right-hand columns of the cash flow table (the cash flow for each group, $C(j)$, followed by a comma, then the number of consecutive cash flows in that group, $N(j)$) into the program. If you select the ungrouped option, the program will automatically enter 1 as $N(j)$ for every group.

Example 1: Enter the ungrouped cash flow in the investment offered to you by George in the example on page 57. Leave the program running at the end so you can do the next example.

Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :■`	Displays the help prompt.
[Ent/Rev]	Begins running the cash flow series entry routine.
`Ent. or Rev.? (E/R) ■`	Asks if you are going to enter numbers from the keyboard or review a current series in the program.
E [END LINE]	Selects the enter option.
`Grp. or Ungr.? (G/U) ■`	Asks if you are going to enter grouped or ungrouped cash flows.
U [END LINE]	Selects the ungrouped option.
`Enter # of periods ■`	Prompts for the number of periods.
6 [END LINE]	Enters the number of periods.
`Enter C(0) ■`	Prompts for the initial cash flow.

Input/Result

`-8200` [END LINE]	Enters the initial cash flow.
`Enter C(1) ■`	Prompts for the cash flow in period 1.
`1500` [END LINE]	Enters the first cash flow.
`Enter C(2) ■`	Continue entering the cash flows for periods 2 through 6.
`1800` [END LINE]	
`Enter C(3) ■`	
`2100` [END LINE]	
`Enter C(4) ■`	
`2400` [END LINE]	
`Enter C(5) ■`	
`2700` [END LINE]	
`Enter C(6) ■`	
`3000` [END LINE]	
`Cash flows entered :■`	Indicates the end of the cash flow entry sequence. The TVM program is running and the keyboard is active.

The same key, [Ent/Rev], permits you to review and edit the current cash flow series. While you review the cash flows you may change any cash flow or the number of times it occurs.

Example 2: Review the cash flow series in the investment offered by George on page 57. The procedure continues directly from the previous example. Leave the program running at the end so that you can move directly to the next example.

Input/Result

`Cash flows entered :█`	Display from previous procedure.
[Ent/Rev]	
`Ent. or Rev.? (E/R) █`	Asks which option you want to use.
R [END LINE]	Selects the review option.

The program first displays the number of groups in addition to the initial group. When you press [END LINE], the program will display the amount of the initial cash flow $C(0)$. When you press [END LINE] again, the number of cash flows in group $N(0)$ is displayed. You continue pressing [END LINE] to view the remaining cash flows. At each step, the variable can be changed by typing over the number displayed. In this way you are able to correct or change entries as desired. You can even enter a new number for the number of groups, as long as the new number is smaller than the current number of groups. By changing the number of groups you can shorten a cash flow series, enter different values, and evaluate the investment as a shorter term investment without reentering the data.

Input/Result

`# of groups = 6`	Displays the number of groups past the initial group.
[END LINE]	
`C(0)= -8200.00`	Displays the initial cash flow.
[END LINE]	
`N(0)= 1.00`	The initial cash flow occurs one time.

Input/Result

[END LINE]

```
C(1)= 1500.00
```

Displays the cash flow in group 1.

[END LINE]

```
N(1)= 1.00
```

The cash flow occurs one time. Continue pressing [END LINE] to review the cash flows.

⋮

[END LINE]

```
C(6)= 3000.00
```

Displays the cash flow in group 6.

[END LINE]

```
N(6)= 1.00
```

The cash flow occurs one time.

[END LINE]

```
End of review :█
```

End of review. The program is running and the keyboard is active.

Let the program continue to run so that you can perform the next exercise.

Writing the Current Cash Flow Series to a File

Use the [Write] key to write the current cash flow series to a data file in memory.

When you press [Write], the program asks you to enter a name. The name you use must follow the file naming conventions described in section 6 of the *HP-71 Owner's Manual*. When the program writes the file to memory, one line for each group of cash flows will be displayed at the current DELAY rate. The first number in the line is the group number, the second is the cash flow, and the third is the number of times the cash flow occurs.

Example: Write the current cash flow series to an HP-71 data file named GEORGE.

The procedure assumes that you have continued directly from the previous example.

Input/Result

`End of review :■`	Display from previous procedure.
[Write]	Begins the routine to write to a data file.
`Write to what file? ■`	Asks you to give the file a name.
GEORGE [END LINE]	Enters the name GEORGE. You can enter either lower- or uppercase letters.

`6 groups`
`0 -8200.00 1.00`
`1 1500.00 1.00`
`2 1800.00 1.00`
`3 2100.00 1.00`
`4 2400.00 1.00`
`5 2700.00 1.00`
`6 3000.00 1.00`
`Done :■`
	Briefly displays each group as it is stored into the data file GEORGE.
	The file has been written, the TVM program is running, and the keyboard is active.

[Exit]	Exits the program.
`TVM ended`	

You now have a data file called GEORGE in the HP-71 memory. The file contains the uneven cash flow series you entered into the machine using the keyboard. You have exited the TVM program and can turn the HP-71 off. You will be able to read that same cash flow file into the TVM program again at any time. Or, if you have a mass storage device such as the HP 82400A Card Reader, you can store the data file onto a magnetic card.

Reading a Cash Flow Series from Memory

Use the [Read] key to read a cash flow data file from the HP-71 memory into the TVM program. The routine prompts you for the name of the file.

Example: We assume that you have a cash flow file named GEORGE in the HP-71 memory, and that you exited the program or turned the machine off after creating GEORGE. Read the cash flow data file named GEORGE into the TVM program as the current file.

Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :■`	Displays help prompt.
[Read]	Begins routine to read a cash flow data file.
`Read what file? ■`	Asks for file name.
GEORGE [END LINE]	Enters the file name.
`GEORGE DATA 161`	Briefly shows the HP-71 data file catalog entry for the file (refer to section 6 of the *HP-71 Owner's Manual*).

`6 groups`
`0 -8200.00 1.00`
`1 1500.00 1.00`
`2 1800.00 1.00`
`3 2100.00 1.00`
`4 2400.00 1.00`
`5 2700.00 1.00`
`6 3000.00 1.00`
	The number of groups past the initial group is displayed. Then the group number, the cash flow for that group, $C(j)$, and the number of times the cash flow occurs, $N(j)$, are displayed at the current DELAY rate for each group as the file is read from memory into the program's current file.

`Done :■`	Indicates that the read routine is complete. The program is still running and the keyboard is active. Leave the program running to do the next example.

The cash flow file is now the current file in the TVM program. You may now use any of the other TVM routines, such as editing the cash flow data, writing the cash flow series to still another file, or calculating *NPV* or *IRR* (which are explained next).

Cash Flow Data File Format

The [Write] routine of the TVM program creates a data file in the HP-71 memory. This file has a specific format. If you write your own BASIC language programs you may wish to study appendix C, "Creating Data Files," to learn the format and how to create and read these data files in your own programs.

Calculating Net Present Value

The [NPV] key begins the routine that calculates and uses the net present value (*NPV*) of the current cash flow series. The complete explanation appears after the introductory exercise below so that you will first be able to run the routine and see what it is doing.

To run the NPV routine, press [NPV]. The program will prompt you to enter an interest rate. The program uses that interest rate and the current cash flow series to calculate three numbers. The numbers are:

- Net present value (*NPV*) of the cash flow series.
- Net equivalent uniform series (*NEUS*) to the *NPV*.
- Net future value (*NFV*) of the cash flow series.

After entering *i%*, you can use [END LINE] to see the numbers displayed consecutively. During the sequence, you can press [Exit] or [ON] to stop at any point.

Example 1: Calculate the *NPV*, the *NEUS*, and the *NFV* of the cash flows for George's investment proposal. Use 14% as the interest rate.

If you are continuing from the previous example, the current cash flow series is the series stored in the HP-71 memory and named GEORGE. If it is not, please do the previous exercises, either entering the cash flow series from the keyboard (beginning on page 58) or reading it into the TVM program from memory (page 63). Let the TVM program continue to run and follow the procedure below.

Input/Result

`Done :█`	Display from the previous example. The data file GEORGE is the current cash flow series in the program.

Input/Result

NPV	Begins running the NPV routine.
`Interest rate %? ■`	Asks you to enter the interest rate.
14 END LINE	Enters the interest rate and begins calculating.
`NPV = 108.32`	Displays net present value.
END LINE	Press END LINE to continue.
`NEUS = 27.86`	Displays the net equivalent uniform series (*NEUS*).
END LINE	Press END LINE to continue.
`NFV = 237.76`	Displays the net future value (*NFV*).
END LINE	Press END LINE to continue.
`Run again? (Y/N) ■`	Asks you if you wish to calculate *NPV* again.
N END LINE	Enters N to stop the routine.
`Done :■`	Let the program continue to run in order to do the next example.

Running the NPV routine is very easy, as you found by doing the example. The interpretation of the results is the subject of many publications in finance literature. A brief explanation of the concept of *NPV* is next. After *NPV*, we'll look at the concepts of *NEUS* and *NFV*.

Two facets of net present value need to be explained. They are

1. The mathematical calculation of *NPV*.
2. The use of *NPV* in finance.

The Mathematical Calculation of *NPV*

Net present value is a number that is the result of a calculation using a series of cash flows (the current cash flow series in the TVM program) and a periodic interest rate. The interest rate is applied to each of the cash flows in the series to calculate its value at time 0 on the time line. The present values of the individual cash flows are added together. That sum is called the present value of the series. The initial cash flow is added to the present value and that sum is called the net present value (*NPV*).

Technical Note: The mathematical formula for *NPV* is the following:

NPV = the cash flow at time 0 +

the cash flow at time 1 divided by $(1+i)^1$ +

the cash flow at time 2 divided by $(1+i)^2$ +

the cash flow at time 3 divided by $(1+i)^3$ + . . . +

the cash flow at time n, the last time period in which there is a cash flow, divided by $(1+i)^n$

where i is the periodic interest rate (expressed as a decimal, not a percent).

The following HP-71 BASIC program uses that formula and will calculate the answer to the *NPV* for the example on page 64. You may find the program useful.

```
10 DATA 14,6,−8200,1500,1800,2100,2400,2700,3000
20 READ I,N,P
30 FOR T = 1 to N @ READ C @ P = P + C/(1 + I/100)^T @ NEXT T
40 DISP P @ END
```

The interest rate, the number of cash flows in addition to the initial cash flow, and the cash flow series are in the DATA statement in line 10. The program reads from the DATA statement and calculates and displays P, the net present value of the series of cash flows.

The TVM program calculates the same answer. However, the TVM program permits you to enter cash flows in groups, rather than individually, for each time period. In order to efficiently use the HP-71's computer capabilities for large numbers of cash flows, the TVM program includes routines that use several complex mathematical formulas that are equivalent to, and give the same answers as, the more simple formula.

The Use of *NPV* in Finance

Net present value (*NPV*) is used by financial decision makers to choose investment projects. If the *NPV* of an investment project is positive, the project is acceptable according to the financial criterion. If the *NPV* is negative, the project is not acceptable.

For a given investment project, *NPV* is the final result of an evaluation process called *discounted cash flow analysis*. Discounted cash flow analysis is the topic of many publications and articles in finance and related professions. We urge you to use publications in your profession as resources to learn the details of the meaning and use (and misuse) of discounted cash flow analysis.

At the base of discounted cash flow analysis are several assumptions about the preferences of persons who own businesses, whether the businesses are corporations, partnerships, or proprietorships, whether large or small. Several of the essential assumptions are:

- The owners want their ownership interest to increase in *value*.
- *Value* is represented by *cash*.
- The owners always prefer $1 now over a promise of $1 later—they have a time preference for money.
- The *NPV* summarizes the owner's financial preferences with respect to the project.

In the literature about discounted cash flow analysis and *NPV*, the interest rate that reflects the owner's time preference for money is called by several different names—*cost of capital*, *discount rate*, and *required rate of return*, are among the more commonly used terms. In this manual, if an example asks that a choice or decision be made, the term *cost of capital* is used. In all other cases, the term *interest rate* is used. You need to refer to the literature to learn the terms that are used in your profession. Likewise, the literature will help you in the problem of choosing that rate *before* you make an *NPV* calculation.

In simplified form, there are four steps to the application of discounted cash flow analysis using *NPV*. They are:

1. Estimate the *incremental* cash flows that will result from the project and summarize them in a cash flow diagram or cash flow table.
2. Choose the interest rate/cost of capital.
3. Calculate *NPV*.
4. Apply the *NPV* decision rule.

The TVM program helps you perform step 3, but you must call on your professional literature and practices, as well as your own practical and educational experience for help in performing the other steps.

The [NPV] key calculates and displays *NPV* and two related numbers: *NEUS* and *NFV*.

The first number displayed is the *NPV* of the current cash flow series in the TVM program. When you press [NPV] the program prompts you to enter the periodic interest rate. If you are going to use the calculated *NPV* in making discounted cash flow decisions, the interest rate that you enter should be your *cost of capital.*

After *NPV* is calculated, press [END LINE] and the program calculates the *net equivalent uniform series* (*NEUS*) to the *NPV*. The *NEUS* is the uniform periodic payment that, when paid at the end of each period over the term of the investment, is equivalent to the *NPV*.

After *NEUS* is calculated, press [END LINE] and the program calculates the *net future value* (*NFV*) of the series of cash flows. The *NFV* is the net amount at the *end* of the investment term that is equivalent to the *NPV*.

The [NPV] key begins the sequence of calculations that results in *NPV*, *NEUS*, and *NFV*. [Exit] or [ON] will exit the sequence at any point.

After the *NFV* is calculated and displayed, press [END LINE] and the prompt asks if you wish to do another calculation.

Let's return to the example on page 57 in which George offered you an investment for $8,200. The *NPV* at 14% is $108.32. If your cost of capital is 14%, the investment is attractive because the exchange of your $8,200 for George's promise would increase your present value by a net amount of $108.32. A typical extension of the use of *NPV* is that you would be willing to pay up to $8,200 + $108.32 for the investment.

The *NEUS* is $27.86, which means that a uniform series of six payments of $27.86 paid at the end of each year is equivalent to a present value of $108.32 if the interest rate is 14%. A typical interpretation of the *NEUS* is that each cash flow from the project could be lowered by up to $27.86 and the investment would still be attractive.

The *NFV* is $237.76, which is the amount to which $108.32 would accumulate at 14% interest compounded annually for six years. A typical interpretation of the *NFV* is that the terminal value of the project could be up to $237.76 lower and still be attractive.

Technical Note: You may get a better understanding of the meaning of *NEUS* and *NFV* if you use the uniform series of cash flows routines to supplement this example. Enter 6 as *n*, 14 as *i%*, and −108.32 as *PV*. Then enter 0 as *FV* and solve for *PMT*. The answer will be equal to the 27.86 *NEUS* figure. If you enter 0 as *PMT* and solve for *FV*, the answer will be equal to the 237.76 *NFV* figure.

Example 2: Continue from the previous example using the data for George's proposal. Calculate *NPV*, *NEUS*, and *NFV* at 15% interest.

This procedure assumes that you have the cash flow series `GEORGE` in the `TVM` program as the current cash flow file.

Input/Result

Input/Result	
`Done :■`	Display from previous example.
[NPV]	Begins running the `NPV` routine.
`Interest rate %? ■`	Asks you to enter the interest rate.
`15` [END LINE]	Enters the interest rate.
`NPV = -142.24`	Displays the *NPV*.
[END LINE]	Press [END LINE] to continue.
`NEUS = -37.59`	Displays the *NEUS*.
[END LINE]	Press [END LINE] to continue.
`NFV = -329.01`	Displays the *NFV*.
[END LINE]	Press [END LINE] to continue.
`Run again? (Y/N) ■`	Asks if you wish to do further calculations.
`N` [END LINE]	
`Done :■`	The procedure is finished, the `TVM` program is running, and the keyboard is active.

Leave the program running so you can work the next example without reentering or restarting the `TVM` program.

Compare the results of example 1 and example 2. You will notice that the *NPV* is lower at the higher interest rate. This is typically the case if the problem is set up from the viewpoint of the investor.*

This means that as your alternative interest rates and costs of capital go up, the present value of any given investment goes down. In these two examples, changing the interest rate from 14% to 15% not only lowers the *NPV*, but changes it from positive to negative! If your cost of capital is 14%, the investment would be attractive. If your cost of capital is 15%, the investment is *not* attractive.

Calculating Internal Rate of Return

The [IRR] key begins a routine that calculates the internal rate of return (*IRR*) of the current cash flow series in the TVM program. We will first calculate the *IRR*, then explain the meaning of the result.

Example 1: Calculate the *IRR* of George's investment proposal, which is the uneven cash flow series used in the last several examples. We will set the display setting to four decimal places.

The procedure that follows assumes that the previous example has just been solved. If this is not the case, please enter the cash flow series from the example on page 58 or read the cash flow file name GEORGE into the TVM program (refer to page 63).

Input/Result

Input/Result	Description
`Done :■`	Display from previous example.
[#Dec]	Begins routine to set number of decimals.
`Enter # of decimals ■`	Asks for number of decimal places.
4 [END LINE]	Sets decimals to four places.
`4.0000 :■`	Display shows four decimal places.

* If you are working on an investment problem for which the *NPV* does not decrease as the interest rate increases, you may have a multiple *IRR* problem such as that described on page 77. If this occurs you should read those pages before completing your analysis of the problem.

Input/Result

[IRR] Begins running the IRR routine.

```
Calculating
```

The calculation of *IRR* takes several seconds.

```
IRR% =   14.4266 :■
```

The *IRR* is displayed.

[Exit] Exits the program.

```
TVM ended
```

The *IRR* of George's investment proposal is 14.4266%. (There are trailing decimals that are not displayed.)

The internal rate of return is defined as that interest rate at which the net present value (*NPV*) equals zero. In an earlier example (page 64) you calculated the *NPV* for George's offer. At 14% interest rate the *NPV* is $108.32. At 15% the *NPV* is $−142.24. At some interest rate between 14% and 15% the *NPV* = 0. That interest rate is the *IRR* of the investment proposal. If you use the *IRR* from this example as the interest rate to calculate *NPV* you will get a number that is very close to zero. If you use 14.4266%, you will get 0.0035.

There is no method to directly calculate the internal rate of return. All computers and calculators that calculate *IRR* use some method of "successive iterations" that try an interest rate, test to see if the *NPV* is close to zero, try another, test the *NPV*, and so on. The iterative method always includes a very small tolerance around zero at which the computer or calculator program will stop and, in effect, say "this is close enough; I will stop calculating and display that interest rate as the *IRR*." That process is the reason that the calculation of *IRR* takes longer than other time value of money calculations.

The *IRR* is used in finance to make decisions about capital investments. To understand how it is used, we need to define the term *conventional investment*. If the cash flow series is structured from the viewpoint of the investor, a conventional investment is one in which all three of the following criteria are met:

- The first cash flows are negative and some cash flows come later that are positive.
- The sequence of cash flows changes sign (negative to positive) only once.
- The sum of all of the cash flows is positive.

If an investment proposal meets this definition, there is one and only one *IRR*.

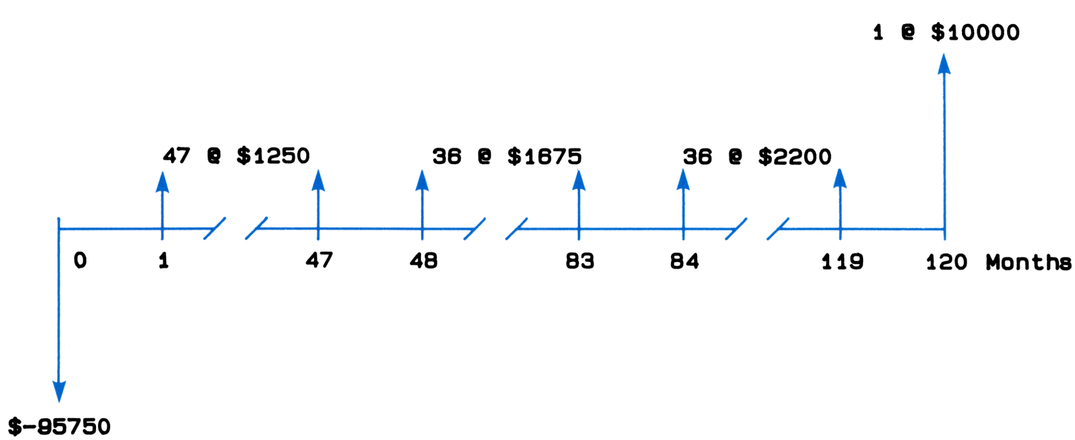
If there is one and only one *IRR*, a powerful decision rule can be applied: if the *IRR* is greater than the investor's cost of capital, the investment is attractive. If the *IRR* is less than the investor's cost of capital the investment is not attractive. For George's proposal, the project is attractive to you if your cost of capital is less than 14.4266%. If your cost of capital is greater than the 14.4266% the project is not attractive.

For investments other than conventional investments, the decision rule may not apply. For the *nonconventional* investment (one that does not meet the three criteria) there are many possible situations. In some cases there is more than one positive internal rate of return!

The *IRR* (as well as the *NPV*) in discounted cash flow analysis is a very powerful tool. As with all powerful tools, you must be careful when you apply it to real problems. Selecting the interest rate to be used as the cost of capital, the effect of reinvesting cash flows, as well as the use of *IRR* when the projected cash flows change sign several times, are all technical questions that are the subjects of many books and articles. We must refer you to the practices and literature in your profession and industry to learn the applications and conventions used there.

Example 2: What is the implicit rate of return on a 120 month lease if the investment value is $97,000 and the lease payments are paid at the *beginning* of each month? The payment schedule is $1,250 for the first 48 payments, $1,675 for the next 36 payments, and $2,200 for the remaining 36 payments. There is a salvage value estimated to be $10,000 at the end of the 120 months. Consider this amount to be a cash flow at the end of the lease. There are no other cash flows to be considered.

The *IRR* calculation is an uneven cash flow problem in which all cash flows are presumed to occur at the end of the period. Since the lease payments will be made at the beginning of each month, we will make an adjustment that forces all of the payments to be at the end of the month. Reduce the initial investment by the amount of the first payment. Then view the remaining payments as end-of-month payments: 47 at $1,250, 36 at $1,675, 36 at $2,200, and 1 at $10,000. From the viewpoint of the investor, the cash flow diagram and the cash flow table are as follows:



j	t	$C(j)$	$N(j)$
0	0	$−95750	1
1	1-47	$1250	47
2	48-83	$1675	36
3	84-119	$2200	36
4	120	$10000	1

Since this series of cash flows is a new cash flow series to the TVM program, you must enter the series using the keyboard. Use the grouped cash flow option. To use the grouped cash flow option, respond to the prompt that the series is to be grouped. The next prompt asks for the number of groups. Please note that this is the number of groups in addition to group $j = 0$. In this example there are four groups in addition to the initial cash flow.

After you enter the number of groups, the prompt will ask you to enter *both* the cash flow $C(j)$ and the number of occurrences $N(j)$, which you enter from the keyboard. When you enter the two numbers, *separate them with a comma*. Each group is entered in succession until the program prompts that the entire cash flow series has been entered.

Input/Result

RUN TVM [END LINE]	Begins running the program.
`For help, press H :■`	Displays the help prompt.
[Ent/Rev]	Begins the cash flow entry routine.
`Ent. or Rev.? (E/R) ■`	Asks you to enter E or R.
E [END LINE]	
`Grp. or Ungr.? (G/U) ■`	Asks you to select grouped or ungrouped (G or U).

Input/Result

G [END LINE]	Selects grouped.
`Enter # of groups ■`	Asks for the number of groups past the initial cash flow.
4 [END LINE]	Enters the number of groups.
`Enter C(0),N(0) ■`	Asks you to enter the cash flow and the number of occurrences for group 0, the initial group. Be sure to separate the two numbers with a comma.
-95750,1 [END LINE]	Enters the negative investment, a comma, and the number of times the cash flow occurs.
`Enter C(1),N(1) ■`	Asks you to enter the information for group 1.
1250,47 [END LINE]	Enters the cash flow and the number of occurrences.
`Enter C(2),N(2) ■`	Asks you to enter the information for group j = 2.
1675,36 [END LINE]	Enters the cash flow and the number of occurrences.
`Enter C(3),N(3) ■`	Asks you to enter the information for group j = 3.
2200,36 [END LINE]	Enters the cash flow and the number of occurrences.
`Enter C(4),N(4) ■`	Asks you to enter the information for group j = 4.
10000,1 [END LINE]	Enters the cash flow and the number of occurrences for the last group.
`Cash flows entered :■`	Indicates that the cash flow series is entered.

At this point you have several options: you can review the cash flow series to see that it was entered correctly (refer to the procedure on page 60), you can write the cash flow series to the HP-71 memory (refer to the procedure on page 61), or you can calculate *IRR* or *NPV*. For this exercise the procedure will continue by calculating the *IRR*. Since the answer will be a periodic *IRR*, the procedure will show you how to perform arithmetic on the result to arrive at an annual *IRR*. The calculated *IRR* is stored as *i*%. You can recall it by pressing [Recall][i%] and do extended arithmetic by using the [Arith] function.

Input/Result

`Cash flows entered :■`	Display from last step in previous procedure.
[IRR]	Begins IRR calculation routine.
`Calculating` `IRR% = 1.26 :■`	Calculates and displays the monthly *IRR*%.
*12 [Arith]	Multiplies the result times 12 to calculate the annual interest rate.
`i% *12 = 15.07 :■`	The annual interest rate. The TVM program is running and the keyboard is active.

Because of the particular procedure used, the TVM program still has the lease example cash flows as its current cash flow file. At this time you can use any of the other TVM functions, including [Ent/Rev] to review and edit the cash flow series, the [IRR] or [NPV] functions, or [Write] to write the cash flow series to a file in memory. To demonstrate this, follow the procedure below. The procedure writes the cash flow series to a file called LEASE (refer to page 61 for the explanation of how the cash flow series are named and written to files).

Input/Result

`i% *12 = 15.07 :■`	Display from previous procedure.
[Write]	Begins the routine to write the current cash flow to a file in memory.
`Write to what file? ■`	Asks for the file name.

Input/Result

LEASE [END LINE]	Enters the name.

```
4 groups
0 -95750.00   1.00
1   1250.00  47.00
2   1675.00  36.00
3   2200.00  36.00
4  10000.00   1.00
Done :■
```

Program displays the cash flow series as it is written to the file.

Completed. The TVM program is running and the keyboard is active.

[Exit]	Exits the program.

```
TVM ended
```

The discussion of uneven cash flows explains the similarities between a uniform series of cash flow and an uneven series of cash flows (refer to page 58). If you use the [IRR] routine frequently, you need to be aware that the program steps that solve for *IRR* are the same as those that solve for *i%*.

When you use the [Ent/Rev] or [Read] keys, you are putting a cash flow series into the TVM program as the current uneven series of cash flows. When you press [IRR], the solution is stored as *i%*.

When you press [i%], the program examines the values of *n*, *PV*, *PMT*, *FV*, and the payment mode, copies that cash flow information into the current uneven cash flow series, then runs the routine to calculate *IRR*. When the solution is found, the program stores and displays the answers as *i%*. If you solve for *i%* after entering *n*, *PV*, *PMT*, and *FV*, the current uneven series of cash flows becomes a series for which the *IRR* can be determined by pressing [IRR].

Multiple IRRs

Technical Note 1: The uneven cash flow series permits calculation of the *IRR* in the case of investments that are *not* conventional (refer to pages 24 and 71). The following example has three *IRR*s.

Example: An investment has the following series of cash flows. Calculate the *IRR*s of the investment.

Group *j*	Time Periods *t*	Cash Flow *C(j)*	Number of Years *N(j)*
0	0	−180	1
1	1-5	100	5
2	6-10	−100	5
3	11-19	0	9
4	20	200	1

The following solution assumes that the cash flow series is the current cash flow series in the TVM program, that the TVM program is running, and that the keyboard is active. (To enter the series of cash flows, follow the procedure described on page 58.)

Input/Result

```
Cash flows entered :■
```

The cash flows are entered, TVM is running, and the keyboard is active.

[IRR] Begins IRR routine.

```
Enter est. of i% ■
```

Prompts you to enter an estimate of *IRR*. Try various numbers and see what happens.

0 [END LINE] Enters 0 as an estimate of *IRR*.

```
Calculating
IRR% =   1.86 :■
```

Program is working.
An *IRR* is found.

[IRR] Begins the *IRR* routine again.

```
Enter est. of i% ■
```

Prompts for an estimate of *IRR*.

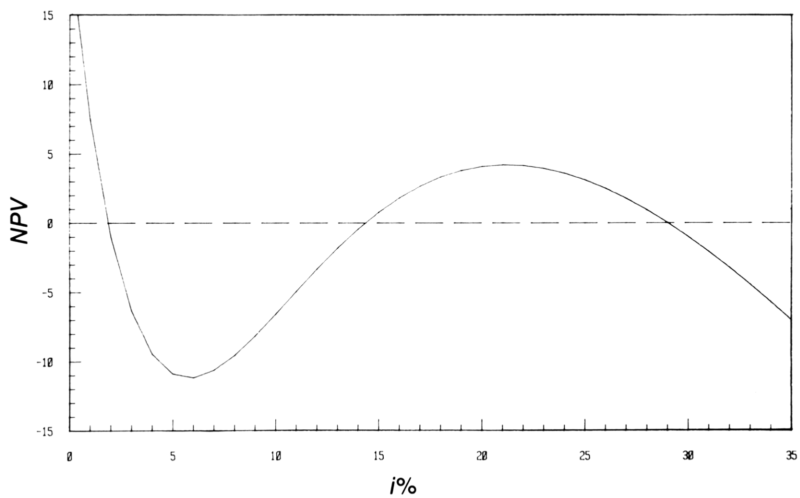
Input/Result

10 [END LINE]	Enters 10% as an estimate of *IRR*.
`Calculating` `IRR% = 14.35 :■`	Another *IRR* is found.
[IRR]	Begins the *IRR* routine again.
`Enter est. of i% ■`	Prompts for an estimate of *IRR*.
50 [END LINE]	Enters 50% as an estimate of *IRR*.
`Calculating` `IRR% = 29.02 :■`	Another *IRR* is found.
[Exit]	Exits the program.
`TVM ended`	

The program calculated the three positive *IRR*s that exist for the series of cash flows in the example.

We must refer you to the literature of your profession for a discussion of the multiple *IRR* situation. In real estate, for example, the financial management rate of return and the modified internal rate of return are two alternatives that are sometimes used as decision models in the multiple *IRR* case.

In financial theory the discounted cash flow model is quite clear: the objective is to maximize value. The decision rule using *NPV* still applies: if the *NPV* of the project is positive at the cost of capital, the project is attractive. To evaluate the situation with multiple *IRR*s, it is helpful to graph the relationship between *NPV* and *i*%. Recall from page 71 that, for an investment with a *conventional* series of cash flows, *NPV* decreases as *i*% increases. But the investment in the multiple *IRR* example is *not* a conventional series. Run the *NPV* routine for this cash flow series. Do so at several different interest rates to see how *NPV* changes as *i*% changes. If you graph the results you will have a picture that looks like the following:



***NPV* as a Function of *i*%**

In the graph the three *IRR*s are at points where the curve crosses the *i*%-axis. Below *i*% = 1.86, and between 14.35 and 29.02, *NPV* is positive. At other interest rates, *NPV* is negative. Strict application of the discounted cash flow model would indicate that the project would be acceptable if *NPV* is positive at the firm's cost of capital.

Technical Note 2: An unconventional series may or may not have a unique positive *i*% or *IRR* solution. The TVM program will detect the unique positive solution. However, this does not mean that negative solutions do not exist. If an unconventional series happens to have both a unique positive solution and also negative solutions, the TVM program may find a negative solution in the iterative search process before it finds the positive one.

If TVM does find a negative *i%* or *IRR*, you can instruct the program to begin its search for a positive solution with your own first estimate of the solution. To do so, interrupt the program by pressing [ON], type I4 =, followed by your estimate of the positive *IRR*, press [END LINE], and then press [f][CONT]. Then solve for *i%* or *IRR* as before.

The TVM program performs a maximum of 15 iterations in its search for an *i%* or *IRR* solution. If one is not found, the Can't solve message is displayed. If you wish to have the program make more than 15 iterations, you can interrupt TVM by pressing [ON]. Then type X4 =, followed by the number of iterations that you desire, press [END LINE], and then press [f][CONT]. Then proceed to solve for *i%* or *IRR*.

Section 5

Depreciation

Introduction

Tangible assets such as buildings, machines, tools, trucks, etc. gradually decline in value through usage, technological obsolescence, environmental deterioration, or a combination of all of these. Depreciation is a way of periodically accounting for the declining value of the asset. In addition, income tax regulations permit these expenses to be used to reduce income in calculating income tax liability.

The program in the Finance Pac that handles depreciation is called DEP. DEP calculates, displays, prints, and creates files of depreciation charges for five different depreciation methods:

- Accelerated Cost Recovery System (AC).
- Straight-line depreciation (SL).
- Declining-balance depreciation (DB).
- Declining-balance depreciation with switchover to straight-line (SW).
- Sum-of-the-years-digits depreciation (SY).

Although DEP is part of the Finance Pac, it is separate from the TVM program. You access the DEP program by typing RUN DEP [END LINE]. It has its own series of prompts that lead you through the program.

Depreciation Calculations

Depreciation that is charged against income is not a cash outlay. However, the amount that income is reduced for tax purposes reduces tax liability, generating a positive cash flow. You are actually paying less taxes than you would if the depreciation were not charged against income. The tax savings that is created is a cash flow when evaluating capital projects.

Several options are available in the DEP program. You can:

- Display the depreciation schedule. You see the schedule by selecting the D (Display) option. You press [END LINE] to view each successive item in the schedule. Pressing [Exit] (the [N] key) or [ON] during the sequence exits the sequence.

- Write a data file into the HP-71 memory. In this case the file is in the form of a cash flow file that can be read into the TVM program for further processing. You write the depreciation charges to a data file by selecting the F (File) option.
- Print a depreciation table (if you have an HP 82401A HP-IL Interface Module and a printer operating in the HP-IL loop). You select the P (Print) option to print a table. You will see the print option only if a printer is attached.
- Exit the program. To do so, select the Q (Quit) option.

Whenever you type RUN DEP [END LINE], the program will progress through three stages in sequence:

1. A sequence of prompts asks you to enter the financial information about the asset.
2. The program asks you to select the depreciation method.
3. The program asks you to select a method of presenting the results. This step is repeated until you type Q to exit from this step.

Example: Calculate the declining-balance depreciation for the following asset and create a data file named ASSET that contains the depreciation charges. The cost of the asset is $15,000. The asset has a 7-year useful life, zero salvage value, and declining-balance depreciation will be taken at 150% of straight-line (a declining-balance factor of 1.5). After the data file is created, display the schedule of depreciation charges. (The procedure assumes that a printer is not attached to the HP-71.)

Input/Result

RUN DEP [END LINE]	Begins running the program.
`Cost of asset? ■`	Prompts for cost basis.
15000 [END LINE]	Enters the cost basis.
`Asset life? ■`	Prompts for asset life.
7 [END LINE]	Enters useful life.
`Salvage value? ■`	Prompts for salvage value.
0 [END LINE]	Enters salvage value.
`AC,SL,DB,SW, or SY? ■`	Asks for depreciation method.

Input/Result

DB [END LINE]	Enters declining-balance as the method.
`DB Factor? (1 to 2) ■`	Prompts for the factor.
1.5 [END LINE]	Enters declining-balance factor of 1.5.

`Calculating`
`What output form?`
`D, F, or Q? ■`
	Calculations have been made and the program asks you to select a method of presenting the results. (If a printer is attached, the prompt will include a P option.)

D [END LINE]	Selects the option to display the schedule.
`Year 1`	Displays the year number.
[END LINE]	Displays the amount of depreciation.
`Deprec. = 3214.29`	
[END LINE]	Displays the remaining book value.
`Book val. = 11785.71`	
[END LINE]	Continue pressing [END LINE] to scroll through the output.
`Year 2`	You can exit from the sequence of displays at any time by pressing [N] or [ON]. (The program skips to the Done statement and continues from there.)
⋮	
`Year 7`	

Input/Result

END LINE

```
Deprec. =  3529.22
```

END LINE

```
Book val. =  0.00
```

END LINE

```
Done
```

```
What output form?
D, F, or Q? ■
```

Asks you to select the method of presentation.

F END LINE

Selects the option to create a memory file.

```
File name? ■
```

Asks you to enter the file name.

ASSET END LINE

Enters ASSET as the name.

```
 7.00 years
0 0.00  1.00
1 3214.29  1.00
2 2525.51  1.00
3 1984.33  1.00
4 1559.12  1.00
5 1225.02  1.00
6 962.52  1.00
7 3529.22  1.00
Done
What output form?
D, F, or Q? ■
```

The file is written in the cash flow format of the TVM program (refer to page 61). The lines of the file (the year number, amount of depreciation, and the number 1) are displayed as they are written to the file.

Asks you to select a method of presentation.

Input/Result

Q END LINE — Selects the Quit option.

```
DEP ended
```

You now have a file in memory named ASSET and have viewed the depreciation schedule. ASSET includes the schedule of depreciation for each year. It is also in the format of a TVM cash flow file (refer to page 64). Notice that each year's remaining book value is not in the file, since it is not part of the flow. ASSET can be read into the TVM program by using the Read key (refer to page 63).

Accelerated Cost Recovery System

The Economic Recovery Tax Act of 1981 made sweeping changes in the tax accounting procedures used to charge the cost of assets against income. Basically, ACRS charges replace the traditional methods of calculating depreciation. The ACRS charges are calculated from percentage tables supplied by the Internal Revenue Service. Refer to their publications to learn the details of the different tables that are used. The example in this section uses the table percentages for an asset in the 5-year class placed in service from 1981 through 1984.

The procedure for running the AC option is different from the other methods that are available in only one respect. The AC option prompts you to enter the ACRS percentages. Other than prompts for these percentages, the prompts are the same as for the other methods. These percentages must sum to 100% over the life of the asset. Always use zero salvage value in calculating ACRS charges.

Example: Calculate and display the annual ACRS charges for an asset in the 5-year class. The cost of the asset is $15,000. A table of the recovery percentages appears below.

Year	Recovery Percentage
1	15
2	22
3	21
4	21
5	21

Input/Result

RUN DEP END LINE — Selects the depreciation program.

```
Cost of asset? ▮
```

Prompts you to enter the asset cost.

Input/Result

15000 END LINE	Enters the asset cost.
`Asset life? ■`	Prompts you to enter the life of the asset.
5 END LINE	Enters the life of the asset.
`Salvage value? ■`	Prompts you to enter the salvage value.
0 END LINE	Enters zero as the salvage value.
`AC,SL,DB,SW, or SY? ■`	Asks you to select a depreciation method.
AC END LINE	Selects accelerated cost recovery.
`Yr. 1 (100% left)? ■`	Prompts for year 1 ACRS %. (The display also shows the total remaining percentage to be entered over the asset life.)
15 END LINE	Enters first year ACRS %.
`Yr. 2 (85% left)? ■`	Prompts for year 2 ACRS %.
22 END LINE	Enters second year ACRS %.
`Yr. 3 (63% left)? ■`	Prompts for year 3 ACRS %.
21 END LINE	Enters third year ACRS %.
`Yr. 4 (42% left)? ■`	Prompts for year 4 ACRS %.
21 END LINE	Enters fourth year ACRS %.
`Yr. 5 (21% left)? ■`	Prompts for year 5 ACRS %.

Input/Result

21 END LINE — Enters fifth year ACRS %.

```
Calculating
What output form?
D, F, or Q? ■
```

Calculations have been made and the program asks you to select a method of presentation.

D END LINE — Selects the display option.

```
Year 1
```

Displays the year number.

END LINE

```
Deprec. =  2250.00
```

Displays the ACRS charge.

END LINE

```
Book val. =  12750.00
```

Displays the remaining book value.

END LINE

```
Year 2
```

Continue pressing END LINE to scroll through the remaining output.

⋮

END LINE

```
Year 5
```

END LINE

```
Deprec. =  3150.00
```

END LINE

```
Book val. =  0.00
```

Input/Result

END LINE

```
Done
```

```
What output form?
D, F, or Q? ■
```

The output selection prompt returns.

Q END LINE

Exits the program.

```
DEP ended
```

Straight-Line Depreciation

The annual straight-line depreciation charge is the difference between the asset cost and the salvage value divided by the life of the asset in years. The charge is the same each year.

You select the straight-line depreciation method by using the SL response when you see the following prompt in the display:

```
AC,SL,DB,SW, or SY? ■
```

Example: What is the schedule of straight-line depreciation charges for an asset that costs $15,000, assuming a 5-year life and zero salvage value?

For the keystroke procedure, refer to the procedure in the explanation of the ACRS method (page 85). (Remember to select the SL option rather than AC.) The printed schedule is the following:

```
Depr. Method: SL
Cost:         15000.00     Life:            5.00     Salvage:          0.00

 Year     Depreciation              Book Value

   0            0.00                 15000.00
   1         3000.00                 12000.00
   2         3000.00                  9000.00
   3         3000.00                  6000.00
   4         3000.00                  3000.00
   5         3000.00                     0.00
```

Declining-Balance Depreciation

Declining-balance depreciation is one of several standard methods in which depreciation charges are greater in the early years of the life of an asset and less in later years. These methods are known as accelerated depreciation methods. The calculation of declining-balance depreciation is related to straight-line depreciation and is often referred to using a percentage of straight-line depreciation, as in "150% of straight-line depreciation." The calculation of the charge for any year is to multiply a factor, 1.5 in the case of 150% declining-balance depreciation, times the remaining depreciable value at the beginning of that year divided by the asset life.

You have the option to select declining-balance depreciation (DB) or declining-balance with switchover to straight-line (SW). The switchover option automatically switches to straight-line depreciation during the year in which the straight-line depreciation charge for that year is greater than the declining-balance charge. You select either of these options by using either the DB or the SW response when you see the following prompt in the display.

AC,SL,DB,SW, or SY? ▒

Example: For an asset with a cost of $15000, a 5-year life, and zero salvage value, calculate two depreciation schedules: the first using the declining-balance method with a factor of 1.5, the second using declining-balance with a factor of 1.5 and automatic switchover to straight-line.

For the details of the keystroke procedure please refer again to the explanation of the ACRS method, page 85. (Select the DB and SW options instead of AC.) The two schedules are as follows:

Depr. Method: 1.50 DB
Cost: 15000.00 Life: 5.00 Salvage: 0.00

Year	Depreciation	Book Value
0	0.00	15000.00
1	4500.00	10500.00
2	3150.00	7350.00
3	2205.00	5145.00
4	1543.50	3601.50
5	3601.50	0.00

Depr. Method: 1.50 SW
Cost: 15000.00 Life: 5.00 Salvage: 0.00

Year	Depreciation	Book Value
0	0.00	15000.00
1	4500.00	10500.00
2	3150.00	7350.00
3	2450.00	4900.00
4	2450.00	2450.00
5	2450.00	0.00

Sum-of-the-Years-Digits Depreciation

Sum-of-the-years-digits depreciation is an accelerated depreciation method that is available to you in DEP. The depreciation charges in the early years of the asset's life are larger than in later years. The calculation is quite simple in concept. As an example, consider an asset with a 5-year life. The base of the calculation is the sum of the numbers 1, 2, and so forth, up through the life of the asset. In this case, $1 + 2 + 3 + 4 + 5 = 15$.

The asset cost basis is divided by 15, and five times this amount is charged as depreciation in the first year, four times the amount is charged in the second year, three times in the third year, two times in the fourth year, and one times in the fifth year.

Example: Calculate the sum-of-the-years-digits depreciation charges for an asset with a cost of $15,000, a 5-year life, and zero salvage value.

For a detailed explanation of the keystroke procedures please refer to the ACRS example (page 85), but use the SY option instead of AC. The printed schedule of the results is the following:

```
Depr. Method: SY
Cost:        15000.00     Life:             5.00     Salvage:          0.00

 Year     Depreciation             Book Value

  0            0.00                15000.00
  1         5000.00                10000.00
  2         4000.00                 6000.00
  3         3000.00                 3000.00
  4         2000.00                 1000.00
  5         1000.00                    0.00
```

Appendix A

Owner's Information

Maintenance

The HP-71 finance module does not require maintenance. However, there are several precautions, listed below, that you should observe.

CAUTIONS

- Do not place fingers, tools, or other objects into the plug-in ports. Damage to plug-in module contacts and the computer internal circuitry may result.
- Turn off the computer (press [f][OFF]) before installing or removing a plug-in module.
- If a module jams when inserted into a port, it may be upside down. Attempting to force it further may result in damage to the computer or the module.
- Handle the plug-in modules very carefully while they are out of the computer. Do not insert any objects in the module connector socket. Always keep a blank module in the computer port when a module is not installed. Failure to observe these cautions may result in damage to the module or the computer.

Limited One-Year Warranty

What We Will Do

The HP 82482A Finance Pac is warranted by Hewlett-Packard against defects in materials and workmanship affecting electronic and mechanical performance, but not software content, for one year from the date of original purchase. If you sell your unit or give it as a gift, the warranty is transferred to the new owner and remains in effect for the original one-year period. During the warranty period, we will repair or, at our option, replace at no charge a product that proves to be defective, provided you return the product, shipping prepaid, to a Hewlett-Packard service center.

What Is Not Covered

This warranty does not apply if the product has been damaged by accident or misuse or as the result of service or modification by other than an authorized Hewlett-Packard service center.

No other express warranty is given. The repair or replacement of a product is your exclusive remedy. **ANY OTHER IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE ONE-YEAR DURATION OF THIS WRITTEN WARRANTY**. Some states, provinces, or countries do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. **IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR CONSEQUENTIAL DAMAGES**. Some states, provinces, or countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights that vary from state to state, province to province, or country to country.

Warranty for Consumer Transactions in the United Kingdom

This warranty shall not apply to consumer transactions and shall not affect the statutory rights of a consumer. In relation to such transactions, the rights and obligations of Seller and Buyer shall be determined by statute.

Obligation to Make Changes

Products are sold on the basis of specifications applicable at the time of manufacture. Hewlett-Packard shall have no obligation to modify or update products once sold.

Warranty Information

If you have any questions concerning this warranty, please contact an authorized Hewlett-Packard dealer or a Hewlett-Packard sales and service office. Should you be unable to contact them, please contact:

- In the United States:

Hewlett-Packard Company
Personal Computer Group
Customer Support
11000 Wolfe Road
Cupertino, CA 95014
Toll-Free Number: (800) FOR-HPPC (800 367-4772)

- In Europe:

Hewlett-Packard S.A.
150, route du Nant d'Avril
P.O. Box CH-1217 Meyrin 2
Geneva
Switzerland
Telephone: (022) 83 81 11

Note: Do *not* send units to this address for repair.

- In other countries:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, California 94304
U.S.A.
Telephone: (415) 857-1501

Note: Do *not* send units to this address for repair.

Service

Service Centers

Hewlett-Packard maintains service centers in most major countries throughout the world. You may have your unit repaired at a Hewlett-Packard service center any time it needs service, whether the unit is under warranty or not. There is a charge for repairs after the one-year warranty period.

Hewlett-Packard products are normally repaired and reshipped within five (5) working days of receipt at any service center. This is an average time and could vary depending upon the time of year and the work load at the service center. The total time you are without your unit will depend largely on the shipping time.

Obtaining Repair Service in the United States

The Hewlett-Packard United States Service Center for battery-powered computational products is located in Corvallis, Oregon:

Hewlett-Packard Company
Service Department
P.O. Box 999
Corvallis, Oregon 97339, U.S.A.
or
1030 N.E. Circle Blvd.
Corvallis, Oregon 97330, U.S.A.
Telephone: (503) 757-2000

Obtaining Repair Service in Europe

Service centers are maintained at the following locations. For countries not listed, contact the dealer where you purchased your unit.

AUSTRIA
HEWLETT-PACKARD Ges.m.b.H.
Kleinrechner-Service
Wagramerstrasse-Lieblgasse 1
A-1220 Wien (Vienna)
Telephone: (0222) 23 65 11

BELGIUM
HEWLETT-PACKARD BELGIUM SA/NV
Woluwedal 100
B-1200 Brussels
Telephone: (02) 762 32 00

DENMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 Birkerod (Copenhagen)
Telephone: (02) 81 66 40

EASTERN EUROPE
Refer to the address listed under Austria.

FINLAND
HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 Espoo 10 (Helsinki)
Telephone: (90) 455 02 11

FRANCE
HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 Les Ulis Cedex
Telephone: (6) 907 78 25

GERMANY
HEWLETT-PACKARD GmbH
Kleinrechner-Service
Vertriebszentrale
Berner Strasse 117
Postfach 560 140
D-6000 Frankfurt 56
Telephone: (611) 50041

ITALY
HEWLETT-PACKARD ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 Cernusco Sul Naviglio (Milan)
Telephone: (2) 90 36 91

NETHERLANDS
HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
NL-1181 KK Amstelveen (Amsterdam)
P.O. Box 667
Telephone: (020) 472021

NORWAY
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 Oesteraas (Oslo)
Telephone: (2) 17 11 80

SPAIN
HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
E-Madrid 16
Telephone: (1) 458 2600

SWEDEN
HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9, Kista
Box 19
S-163 93 Spanga (Stockholm)
Telephone: (08) 750 20 00

SWITZERLAND
HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH-8967 Widen
Telephone: (057) 31 21 11

UNITED KINGDOM
HEWLETT-PACKARD Ltd
King Street Lane
GB-Winnersh, Wokingham
Berkshire RG11 5AR
Telephone: (0734) 784 774

International Service Information

Not all Hewlett-Packard service centers offer service for all models of HP products. However, if you bought your product from an authorized Hewlett-Packard dealer, you can be sure that service is available in the country where you bought it.

If you happen to be outside of the country where you bought your unit, you can contact the local Hewlett-Packard service center to see if service is available for it. If service is unavailable, please ship the unit to the address listed above under "Obtaining Repair Service in the United States." A list of service centers for other countries can be obtained by writing to that address.

All shipping, reimportation arrangements, and customs costs are your responsibility.

Service Repair Charge

There is a standard repair charge for out-of-warranty repairs. The repair charges include all labor and materials. In the United States, the full charge is subject to the customer's local sales tax. In European countries, the full charge is subject to Value Added Tax (VAT) and similar taxes wherever applicable. All such taxes will appear as separate items on invoiced amounts.

Products damaged by accident or misuse are not covered by the fixed repair charges. In these situations, repair charges will be individually determined based on time and materials.

Service Warranty

Any out-of-warranty repairs are warranted against defects in materials and workmanship for a period of 90 days from date of service.

Shipping Instructions

Should your unit require service, return it with the following items:

- A completed Service Card, including a description of the problem.
- A sales receipt or other proof of purchase date if the one-year warranty has not expired.

The product, the Service Card, a brief description of the problem, and (if required) the proof of purchase date should be packaged in adequate protective packaging to prevent in-transit damage. Such damage is not covered by the one-year limited warranty; Hewlett-Packard suggests that you insure the shipment to the service center. The packaged unit should be shipped to the nearest Hewlett-Packard designated collection point or service center. Contact your dealer for assistance. (If you are not in the country where you originally purchased the unit, refer to "International Service Information" above.)

Whether the unit is under warranty or not, it is your responsibility to pay shipping charges for delivery to the Hewlett-Packard service center.

After warranty repairs are completed, the service center returns the unit with postage prepaid. On out-of-warranty repairs in the United States and some other countries, the unit is returned C.O.D. (covering shipping costs and the service charge).

Further Information

Service contracts are not available. Circuitry and designs are proprietary to Hewlett-Packard, and service manuals are not available to customers. Should other problems or questions arise regarding repairs, please call your nearest Hewlett-Packard service center.

When You Need Help

Hewlett-Packard is committed to providing after-sale support to its customers. To this end, our customer support department has established phone numbers that you can call if you have questions about this product.

Product Information. For information about Hewlett-Packard dealers, products, and prices, call the toll-free number below:

(800) FOR-HPPC
(800 367-4772)

Technical Assistance. For technical assistance with your product, call the number below:

(408) 725-2600

For either product information or technical assistance, you can also write to:

Hewlett-Packard
Personal Computer Group
Customer Support
11000 Wolfe Road
Cupertino, CA 95014

Appendix B

Error Messages

When you use the programs in the HP-71 Finance Pac, you may occasionally see messages in the display that are not described in the earlier sections of this manual. These messages indicate that the program cannot recognize or use an item of input. For example, if you enter a plus sign as part of a file name, the program will display a message indicating that the file name is "illegal," and give you the opportunity to select a new name.

Some of the error messages are generated by the HP-71 outside of the Finance Pac. These messages are distinguished by the letter E followed by a number of up to four digits, a colon, and then the message. These are "system" messages and are explained under "Errors, Warnings, and System Messages" in the *HP-71 Reference Manual.*

Other error messages are generated by the Finance Pac programs. This appendix lists and explains these messages. They are presented in alphabetical order.

`All cash flows = 0`

- An *IRR* or *i%* solution is being attempted on a cash flow series where all the cash flows equal zero. Enter the cash flows and solve again.

`B or E please`

- The only responses permitted are the letters B or E. The previous prompt reappears in the display.

`Can't solve`

- The program cannot find a solution to *i%* or *IRR* in 15 iterations. A solution may not exist. Review the cash flows for proper use of the cash flow sign conventions and try again.
- The *n* solution is out of the range of the machine.

`D or P please`

- The only responses permitted are the letters D or P. The previous prompt reappears in the display.

`Duplicate file` followed by `Replace? (Y/N)`

- The program has found another file in memory with the same name. If you wish to replace the file, and erase the previous file, press [Y][END LINE]. Otherwise press [N][END LINE].

`Enter est. of i%`

- An *i%* or *IRR* solution is being attempted on a cash flow series where there may be zero, one, or more positive solutions. (Refer to page 77). Enter an estimate of a solution and press [END LINE].

`E or R please`

- The only responses permitted are the letters `E` or `R`. The previous prompt reappears in the display.

`Full years > 0, please`

- The asset life must be a full number of years; partial depreciation is not permitted. A negative number or zero is not permitted.

`G or U please`

- The only responses permitted are the letters `G` or `U`. The previous prompt reappears in the display.

`Illegal file name`

- A file name has been entered that violates the HP-71 file naming conventions (refer to section 6 of the *HP-71 Owner's Manual*). The program returns to the help prompt and the active keyboard. You can then press [Write] and use another file name.

`Input error`

- The number, expression, or string in the display when an active key is pressed is not recognized by the program. Reenter the value and try again.

`Integer <= n, please`

- When editing a current uneven cash flow series, the number of groups of cash flows cannot be greater than the total number of cash flows (*n*) and the number must be a whole number. The previous prompt reappears in the display.

`i% must be >= 0`

- The `TVM` program does not permit input of *i%* that is less than zero. Reenter and proceed.

`Low memory`

- The available memory is such that an `Insufficient Memory` error message may occur and terminate the `TVM` program. This warning occurs any time you begin running `TVM` when the HP-71 has fewer than about 2000 bytes of free memory available. When you see this message, you may wish to exit the program and catalog the files to see if some can be purged to free more memory.

`Must be integer > 0`

- The number of groups in an uneven cash flow series must be an integer (whole number) that is greater than zero. Reenter the number and proceed.

`Must be > 0, please`

- The number you enter must be greater than zero.

`Must be >= 0, please`

- The number you enter must be greater than or equal to zero.

`n must be > 0`

- The `TVM` program will not accept negative values of n. Reenter a positive value for n and proceed.

`No file named` *name*

- The program is trying to read a cash flow data file that does not currently exist in RAM. Press [ON] to interrupt the program, catalog the RAM files to determine the correct file name, and press [f] [CONT] to try the procedure again.

`No solution > 0`

- An $i\%$ or IRR solution is being attempted for which there is no solution that is larger than zero. Review the cash flows, particularly for the cash flow sign convention, and solve again.
- An n solution that is greater than zero does not exist.

`Not a TVM flow file.`

- The program is attempting to read a file that is not in the format required by the `TVM` program. Press [ON] to interrupt the program, catalog the HP-71 to determine the correct file, and press [f] [CONT] to proceed.

`Not enough memory`

- The HP-71 currently does not have enough memory to work on the uneven cash flow series that is being entered. This error can occur when you are entering cash flows from the keyboard, writing a current cash flow series to a `DATA` file, or reading a `DATA` file into the `TVM` program. Interrupt the program by pressing [ON], catalog the files in RAM, `PURGE` those that are not needed, press [f] [CONT], and repeat the operation.

`Number, comma, number`

- Grouped cash flows must be entered as a number, a comma, then a number—that is, cash flow, comma, number of occurrences of that cash flow.

`Numeric Input`

- The required input is a number. The characters entered are not recognized by the program as a number. Reenter and press [END LINE].

`Printer not available`

- A printer is not available to print a depreciation schedule. Prompt returns to ask `What output form?`.

`String Ovfl`

- There were more than 32 characters in the display when a key was pressed.

`Y or N please.`

- The only responses permitted are the letters `Y` or `N`. The previous prompt reappears in the display.

`45 is maximum, please`

- Asset life must be 45 or fewer years.

Appendix C

Creating Data Files

If you write your own BASIC language programs for the HP-71, you should be familiar with how to construct data files that can be used by the TVM program. Before you continue, make sure that you are familiar with the use of the uneven cash flow series features of the TVM program, described in section 4. Also, the following discussion assumes that you are able to write simple BASIC language programs for the HP-71 and that you are somewhat familiar with section 14 of the *HP-71 Owner's Manual.*

The discussion begins with an exercise that includes a program to create an uneven cash flow data file that can be read by the TVM program using the [Read] routine. Then, the details of the program are explained so that you can write the statements into your own programs. The discussion ends with an exercise that includes a program to read an uneven cash flow data file into your own program.

The Data File Program

Example: Suppose that your job includes the task of finding the capitalized value of a series of lease payments. To do this, you need to calculate the present value of the lease payments at various interest rates. You work with many different leases, but they have several common features. In particular, all of the leases have monthly payments, lease periods that run over a number of years, and the lease payment often changes on annual anniversary dates, as with renewal options. You wish to create an uneven cash flow data file for each of these leases so that the files can be read into the TVM program for calculation of present value.

Example: There is an advance payment of $2000 with monthly payments at the end of each month as follows:

Monthly Payment	# of Years
$ 750	2
1,100	4
1,400	4
2,000	10
2,800	10

Enter the following program into your HP-71 with the file name WRITEOUT. You do not need to enter the comments into your machine.

```
10 INPUT "What file name? "; A$             ! Enter name for the cash flow file.
20 INPUT "# of groups? "; M                 ! Enter the number of groups.
30 CREATE DATA A$, M+4, 16                  ! Creates a data file in RAM.
40 ASSIGN #1 TO A$                          ! Opens a channel to the data file.
50 PRINT #1,0; "HPAFNN"                     ! Writes record 0 to the file.
60 PRINT #1,1; M+1                          ! Writes record 1 to the file.
70 PRINT #1,2; 0                            ! Writes a zero into record 2 to the
                                            ! file.
80 INPUT "Enter advance ";C                 ! Enter the amount of the advance.
90 PRINT #1,3; C,1                          ! Writes initial cash flow (and 1 as
                                            ! the number of occurrences) to
                                            ! record 3.
100 FOR J = 1 TO M
110 DISP "Enter group ";J
120 INPUT "Lease pmt. ";C                   ! Program loops to let you enter the
130 INPUT "# of yrs. ";N                    ! number of successive cash flows
140 PRINT #1,J+3; C, N*12                   ! and number of years for each cash
150 NEXT J                                  ! flow.
160 ASSIGN #1 TO *                          ! Clear the channel to the file.
170 DISP "Done"
180 END                                     ! Ends the program.
```

Use the following procedures for running the program. We will name the data file DEALA.

Input/Result

RUN WRITEOUT [END LINE]	Begins running the program.
`What file name? ■`	Asks you to enter the file name.
DEALA [END LINE]	Names the file.
`# of groups? ■`	Asks you to enter the number of groups.
5 [END LINE]	Enters 5 as the number of groups.
`Enter advance ■`	Asks you to enter the advance.

Input/Result

2000 [END LINE]	Enters the advance.
`Enter group 1.00` `Lease pmt. ■`	Asks you to enter the lease payment for group 1. The number of decimal places depends upon the current display setting.
750 [END LINE]	Enters the lease payment.
`# of yrs. ■`	Asks you to enter the number of years for this payment.
2 [END LINE]	Enters the number of years.
⋮	Continue entering the information for each group.
`Done`	Program ends.

The program has created a data file named DEALA. You can verify its existence by typing CAT DEALA [END LINE].

If you run the TVM program, press [Read], and respond with the file name DEALA, you will see the uneven cash flow series displayed as it is read into the program.

The Data File Format

The data file that is required by the TVM program has the following characteristics:

- Record 0 must be the string HPAFMN, which stands for HP Applications Format.
- Record 1 must be the total number of data records in the file. This will be one plus the number of groups past the initial cash flow.
- Record 2 must be the number 0.
- Record 3, and any subsequent records, include two numbers: the cash flow for the group and the number of occurrences of that cash flow ($C(j)$ and $N(j)$ respectively).

These lines are written to the file in lines 50 through 70, line 90, and line 140 of the program.

Additionally, you must have a CREATE statement in your program if the file does not already exist. The CREATE statement (line 30 in the program) directs the HP-71 to create a data file, named A$, with M+4 records, each of which is 16 bytes long.

The program creates and writes to this file. After DEALA is created, the file would look like this:

Record #	Data in that record
0	HPAFNN
1	6
2	0
3	2000,1
4	750,24
5	1100,48
6	1400,48
7	2000,120
8	2800,120

Reading a Data File Into Your Program

Now that a data file named DEALA exists in the HP-71 memory, you need to know how to read from that file into your own program.

Example: Read the DEALA file to check that the cash flow information is correct. (You could use the TVM program to review the cash flows. However, the following program allows you to read the information into your own programs.)

Key the following program into your HP-71 and name it READOUT. It is not necessary to key in the comments.

```
10 INPUT "What file name? "; A$         ! Enter the file name.
20 ASSIGN #1 TO A$                      ! Opens a channel to the file.
30 READ #1,1; M                         ! Read the number of records in the
                                        ! file.
40 M = M - 1                            ! Calculates the number of groups
                                        ! past the initial group.
50 DISP M;" groups"                     ! Displays the number of groups.
60 FOR J = 0 TO M
70 READ #1, J + 3; C,N                  ! Reads and displays each cash flow
80 DISP C; N                            ! and the number of occurrences.
90 NEXT J
100 ASSIGN #1 TO *                      ! Closes the channel to the file.
110 DISP "Done"
120 END                                 ! Ends the program.
```

After the program is entered, use the following procedure. Be sure to set the display `DELAY` rate to a comfortable rate for you.

Input/Result

`RUN READOUT` [END LINE]	Begins running the program.
`What file name? ■`	Asks you to enter the file name.
`DEALA` [END LINE]	Enters the file name.

`5.00 groups`
`2000.00  1.00`
`750.00  24.00`
`1100.00  48.00`
`1400.00  48.00`
`2000.00  120.00`
`2800.00  120.00`
`Done`
	The number of groups and the cash flow series will be displayed in sequence. The number of decimal places depends upon the current display setting.
	
	The program ends.

To read from a `TVM` uneven series cash flow data file, your program must first read record 1 to learn how many records there are (as in line 30) and calculate the number of groups (as in line 40). Then, the program reads from the successive records, beginning with record 3 (as in the `FOR` ... `NEXT` loop in lines 60 to 90).

Appendix D

File Names

The finance module contains or uses several files, and each file has a name. These names must not be used as the names of files in user memory, as the HP-71 first searches its own memory before searching the plug-in modules. The following list gives the name of each file in, or used by, the finance module, along with a brief description of the file.

`Finance`	A LEX file containing the software version of the finance module. The `VER$` command reads this information.
`TVM`	The time value of money BASIC language program.*
`TVMKEYZ`	The `TVM` keyboard file.
`TVMKEYS`	A temporary copy of `TVMKEYZ` in the HP-71 memory while `TVM` is running.
`USERKEYS`	A temporary file in memory that stores your own user defined keys, if any, while `TVM` is running and the keyboard is active.
`TVMHELP`	A TEXT file that contains the [Help] key messages.
`DEP`	The depreciation BASIC language program.*
`KEYWAIT`	A LEX file containing the keyword `KEYWAIT$`, which is used in `TVM` and `DEP`. When the `KEYWAIT$` function is executed, the HP-71 goes to a low power consumption state until a key is pressed, and then returns the key name. This is similar to `KEY$`.

* `TVM` and `DEP` can be run without a file name conflict by entering `RUN TVM:PORT(`*x*`)`, where *x* is the number of the port that the finance module is plugged into.

Appendix E

Financial Formulas

Uniform Series of Payments

$$0 = PV + PMT \left[\frac{1 - (1 + i)^{-n}}{i}\right] (1 + iM) + FV\,(1 + i)^{-n}$$

where:
n = the number of compounding periods
i = periodic interest rate, expressed as a decimal
PV = present value
PMT = periodic payment
FV = future value
M = 0 in END mode; 1 in BEGin mode

Amortization

$$Int_j = [BAL_{j-1} \times i] \times (\text{sign of } PMT)$$

$$Prn_j = PMT - Int_j$$

$$BAL_j = BAL_{j-1} - Prn_j$$

where:
Int_j = interest portion of the jth payment
Prn_j = principal portion of jth payment
BAL_j = remaining balance after the jth payment
i = interest rate as a decimal

Net Present Value (Ungrouped Cash Flows)

$$NPV = CF_0 + \sum_{t=1}^{n} \frac{CF_t}{(1 + i)^t}$$

where: CF_t = tth cash flow

i = periodic discount rate as a decimal

Net Present Value (Grouped Cash Flows)

$$NPV = \sum_{j=0}^{M} C_j \left[\left(\frac{(1 + i)^{-n_j} - 1}{i} \right) / \, (1 + i)^{N_j - 1} \right]$$

where: $N_j = \sum_{k=0}^{j-1} n_k$

M = number of groups in addition to the initial group

j = group number (j = 0, 1, . . ., M)

C_j = cash flow in group j

n_j = number of periods in which C_j occurs

i = periodic interest rate

Internal Rate of Return

The *IRR* solution uses the *NPV* formula. The *IRR* is the value of i that results in a *NPV* of zero. To find that value of i, the program makes successive iterations with different values of i until the *NPV* is very close or equal to zero.

Straight-Line Depreciation

$$Dep_n = \frac{Cost - Salvage}{Life}$$

Declining-Balance Depreciation

$$Dep_n = B_{n-1} \left(\frac{R}{Life} \right)$$

where: n = year number

B = remaining depreciable amount

R = multiple of straight-line at which the balance is to decline

Sum-of-the-Years-Digits Depreciation

$$Dep_n = \frac{2 \times (Life - n + 1)}{Life \times (Life + 1)} (Cost - Salvage)$$

Subject Index

Page numbers in **bold** type indicate primary references; page numbers in standard type indicate secondary references.

HEWLETT PACKARD

Portable Computer Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

European Headquarters
150, Route Du Nant-D'Avril
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

HP-United Kingdom
(Pinewood)
GB-Nine Mile Ride, Wokingham
Berkshire RG11 3LL

82482-90001 English

Printed in Singapore 12/83-CP