

# TI-86 Financial Functions

<b>Loading and Installing Finance Features on Your TI-86 .....</b>	<b>2</b>
Loading the Finance Features into TI-86 Memory .....	2
Installing the Finance Features for Use .....	2
Displaying the FIN (Finance) Menu .....	3
The FIN Menu .....	3
Uninstalling the Finance Features .....	3
Deleting the Finance Program from TI-86 Memory .....	3
<b>The TVM (Time-Value-of-Money) Variables .....</b>	<b>4</b>
FIN VARS (Finance Variables) Menu .....	4
<b>Setting the Payment Format .....</b>	<b>4</b>
Payment Format Editor .....	4
Entering Cash Inflows and Cash Outflows .....	4
<b>Using the TVM (Time-Value-of-Money) Solver .....</b>	<b>5</b>
FIN TVM Solver Menu .....	5
Solving for an Unknown TVM Variable (Payment Amount) .....	5
Financing a Car .....	6
Computing Compound Interest .....	6
<b>Using the Financial Functions .....</b>	<b>7</b>
Entering Cash Inflows and Cash Outflows .....	7
FIN FUNC (Financial Functions) Menu .....	7
Calculating Time-Value-of-Money .....	7
Calculating Cash Flows .....	8
Calculating Amortization .....	9
Amortization Example: Calculating an Outstanding Loan Balance .....	10
Calculating Interest Conversion .....	11
Finding Days Between Dates .....	11
Defining the Payment Method .....	11
<b>Menu Map for Financial Functions .....</b>	<b>12</b>
MATH Menu (where FIN is automatically placed) .....	12
(MATH) FIN (Financial) Menu .....	12
FIN TVM (Time-Value-of-Money) Solver Menu .....	12
FIN FUNC (Financial Functions) Menu .....	12
FIN VARS (Financial Variables) Menu .....	12
FIN FORMT (Financial Format) Menu .....	12


## Loading and Installing Finance Features on Your TI-86

To load the financial features onto your TI-86, you need a computer and the TI-86 Graph Link software and cable. You also need to download the finance program file from the Internet and save it on your computer.

When sending a program from your computer to the TI-86, the calculator must **not** be in Receive mode. The Receive mode is used when sending programs or data from one calculator to another.

The executable file associated with the assembly language program (**finexe**) appears on the PRGM NAMES menu, but you need not do anything with it.

### Loading the Finance Features into TI-86 Memory

- 1 Start the TI-86 Graph Link on your computer. (WLink86.exe)
- 2 Turn on your TI-86 and display the home screen. **[ON]**  
**[2nd]** **[QUIT]**
- 3 Click on the Send button on the TI-86 Graph Link toolbar to display the Send dialog box. 
- 4 Specify the finance program file as the file you want to send. finance1.86g
- 5 Send the program to the TI-86. The program and its associated executable file become items on the PRGM NAMES menu.
- 6 Exit Graph Link



### Installing the Finance Features for Use

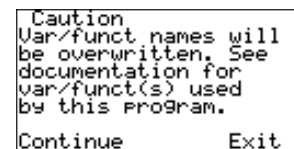
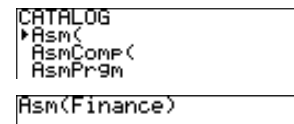
Use the assembly language program **Finance** to install the finance features directly into the TI-86's built-in functions and menus. After installation, the finance features are available each time you turn on the calculator. You do not need to reinstall them each time. When you run assembly language programs that do not install themselves into the **[2nd]** **[MATH]** **[MORE]** menu, their features are lost when you turn off the calculator.

All examples assume that **Finance** is the only assembly language program installed on your TI-86. The position of **FIN** on the **MATH** menu may vary, depending on how many other assembly language programs are installed.

For assembly language programs that must be installed, up to three can be installed at a time (although the TI-86 can store as many as permitted by memory). To install a fourth, you must first uninstall (page 3) one of the others.

The variables that will be overwritten are listed on the **FIN FUNC** and **FIN VARS** menus (page 12).

- 1 Select **Asm(** from the CATALOG to paste it to a blank line on the home screen. **[2nd]** **[CATLG-VARS]** **[F1]**  
**[↓]** (move **▶** to **Asm( )** **[ENTER]**
- 2 Select **Financ** from the PRGM NAMES menu to paste **Finance** to the home screen as an argument. **[PRGM]** **[F1]** (select **Financ)** **[ ]**
- 3 Run the installation program. **[ENTER]**



**Caution:** If you have values stored to variables used by the finance features, they will be overwritten. To save your values, press **[F5]** to exit and then store them to different variables. Then repeat this installation.

- 4 Continue the installation. (Your version number may differ from the one shown in the example.) **[F1]**

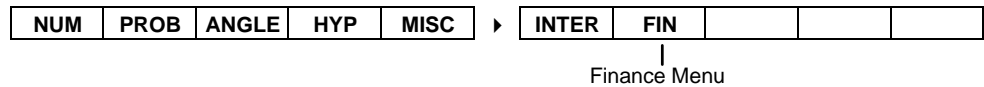
```
var/funct(s) used
by this program.
Continue      Exit
Finance v0.2  Done
```

- 5 Display the home screen. **[CLEAR]**

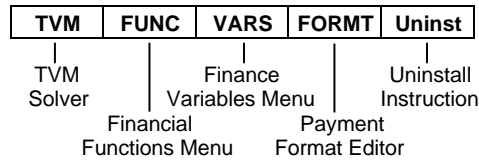
**Displaying the FIN (Finance) Menu** **[2nd] [MATH] [MORE]**

*If other assembly language programs are installed, FIN may be in a menu cell other than [2nd] [MATH] [MORE] [F2].*

When you install the financial program on your TI-86 and activate it, **FIN** becomes the last item on the **MATH** menu.



**The FIN Menu** **[2nd] [MATH] [MORE] [F2]**



**Uninstalling the Finance Features**

*When you uninstall the finance features, the finance assembly language programs (Finance and finexec) remain in memory, but the FIN option is removed from the MATH menu.*

- 1 Display the FIN menu, and then select **Uninst**. **[2nd] [MATH] [MORE] [F2]**  
**[F5]**
- 2 If you are sure you want to uninstall, select **Yes** from the confirmation menu. The FIN menu is removed and the home screen is displayed. (Your version number may differ from the one shown in the example.) **[F4]**

```
TVM  FUNC  VARS  FORMT  Uninst
Uninstall
Finance v0.2
Are you sure?
YES  NO
```

**Deleting the Finance Program from TI-86 Memory**

*Deleting the program does not delete the variables associated with the program.*

- 1 Select **DELET** from the MEM menu. **[2nd] [MEM] [F2]**
- 2 Select **PRGM** from the MEM DELET menu. **[MORE] [F5]**
- 3 Move the selection cursor to **Finance**, and then delete it. **[↓] (as needed) [ENTER]**
- 4 Move the selection cursor to **finexec**, and then delete it. **[↓] (as needed) [ENTER]**

```
RAM  DELET  RESET  TOL  ClrEnt
MATH  STRNG  EQU  CONS  PRGM
DELETE:PRGM
▶Finance  2413 PRGM
  finexec  3110 PRGM
DELETE:PRGM
▶finexec  3110 PRGM
```

## The TVM (Time-Value-of-Money) Variables

**FIN VARS (Finance Variables) Menu**    2nd [MATH] MORE F2 F3

TVM	FUNC	VARS	FORMT	Uninst					
N	I	PV	PMT	FV	▶	PY	CY		

Prompts that correspond to some TVM variables are shown in parentheses.

- N**    Number of payment periods
- I**    Interest rate (I%=)
- PV**    Present value of loan or lease
- PMT**    Payment amount
- FV**    Future value of loan or lease
- PY**    Payments per year (P/Y=)
- CY**    Compounding periods per year(C/Y=)

- ◆ When you enter a value at prompts in the payment format editor (page 4) or the TVM Solver (page 5), the corresponding variable values are updated.
- ◆ When you solve for a TVM variable using the TVM Solver, the corresponding variable value is updated.
- ◆ When you enter numbers as arguments for a TVM function, the corresponding variable values are *not* updated.
- ◆ When you solve for a TVM variable using a TVM function, the corresponding variable value is *not* updated.

## Setting the Payment Format

**Payment Format Editor**    2nd [MATH] MORE F2 F4

TVM	FUNC	VARS	FORMT	Uninst
-----	------	------	-------	--------

The payment format settings define the number of payments per year (P/Y), the number of compounding periods per year (C/Y), and whether the payments are received at the end or beginning of each period (PMT:END BEGIN).



You also can change a setting by storing a value to PY or CY or by executing pEnd or pBegin (page 11).

The payment format editor to the right shows the defaults. To change payments per year or compounding periods per year, enter a new value. To change the payment due setting, move the cursor onto **END** or **BEGIN**, and then press [ENTER].

## Entering Cash Inflows and Cash Outflows

When using the financial functions, you must enter cash inflows (cash received) as positive numbers and cash outflows (cash paid) as negative numbers. The financial functions follow this convention when computing and displaying answers.

## Using the TVM (Time-Value-of-Money) Solver

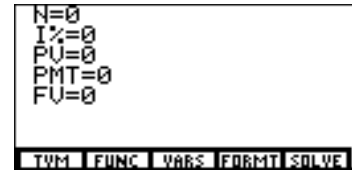
**FIN TVM Solver Menu**    **2nd** **[MATH]** **[MORE]** **[F2]** **[F1]**

**TVM**   **FUNC**   **VARS**   **FORMT**   **SOLVE**

The TVM Solver displays prompts for the five time-value-of-money (TVM) variables.

To solve for an unknown variable, enter the four known variable values, move the cursor to the unknown variable prompt, and then select **SOLVE** (**[F5]**) from the FIN TVM Solver menu. Values

displayed on the TVM Solver are stored to corresponding TVM variables.



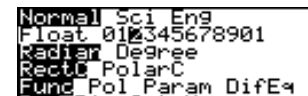
When the TVM Solver is displayed, **SOLVE** replaces **Uninst** on the FIN menu.

### Solving for an Unknown TVM Variable (Payment Amount)

You want to buy a \$100,000 house with a 30-year mortgage. If the annual percentage rate (APR) is 18%, what are the monthly payments?

- Set the fixed-decimal mode to 2 decimal places to display all numbers as dollars and cents.

**2nd** **[MODE]** **▾**  
**▸** **▸** **▸** **[ENTER]**



- Select **FIN** from the MATH menu to display the FIN menu.

**2nd** **[MATH]** **[MORE]** **[F2]**



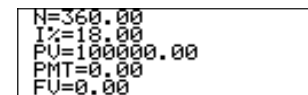
- Select **FORMT** from the FIN menu to display the payment format editor. Set 12 payments per year, 12 compounding periods per year, and payments received at the end of each payment period.

**[F4]** **12** **▾** **12** **▾**  
**[ENTER]**



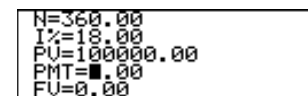
- Display the TVM Solver and enter the known values for four TVM variables. The **N** value of 360 was derived from 30 (years) **[ ]** 12 (months).

**[F1]** **360** **▾** **18** **▾**  
**100000** **▾** **▾** **0**  
**[ENTER]**



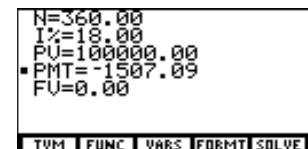
- Move the cursor to the **PMT** TVM variable.

**▴**



- Select **SOLVE** to compute the answer. A small square is displayed next to the solution variable. The answer is stored to the corresponding TVM variable.

**[F5]**



Enter cash inflows as positive numbers and cash outflows as negative numbers.

You cannot leave a variable blank. If you do not have a value, set it to zero.

### Financing a Car

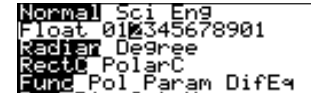
You have found a car you would like to buy. The car costs \$9,000. You can afford payments of \$250 per month for four years. What annual percentage rate (APR) will make it possible for you to afford the car?

When you change **P/Y**, **C/Y** changes automatically.

As you enter a value at any TVM Solver prompt, the corresponding TVM variable value is updated.

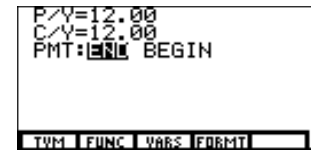
- Set the fixed-decimal mode to 2 decimal places to display all numbers as dollars and cents.

2nd [MODE]  $\downarrow$   
 $\rightarrow \rightarrow \rightarrow$  [ENTER]



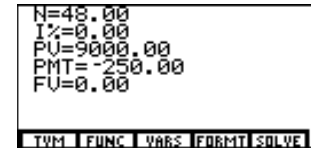
- Display the payment format editor. Set payments per year and compounding periods per year to 12. Set payment due at the end of each period.

2nd [MATH] [MORE] [F2]  
 [F4] 12  $\downarrow$   $\downarrow$  [ENTER]



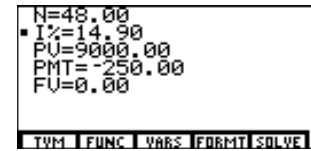
- Display the TVM Solver. Enter 48 monthly payments, present value of \$9,000, payment amount of -\$250 (negation indicates cash outflow), and future value of \$0. The **N** value (48) was derived from 4 (years)  $\times$  12 (months).

[F1] 48  $\downarrow$   $\downarrow$  9000  
 $\downarrow$  (-) 250  $\downarrow$  0  
 [ENTER]



- Move the cursor to **I% =** (interest rate) and then select **SOLVE** from the TVM Solver menu. A small square is displayed next to the solution. The solution value is stored to the TVM variable **I**.

$\uparrow \uparrow \uparrow$  [F5]



### Computing Compound Interest

At what annual interest rate, compounded monthly, will \$1,250 accumulate to \$2,000 in 7 years?

Because there are no payments when you solve compound interest problems, you must set **PMT** to 0 and set **P/Y** to 1.

The decimal mode is fixed at 2 from the previous example.

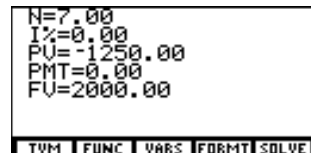
- Display the payment format editor. Set payments per year to 1 and compounding periods per year to 12. Set payment due at the end of each period.

2nd [MATH] [MORE] [F2]  
 [F4] 1  $\downarrow$  12  $\downarrow$   
 [ENTER]



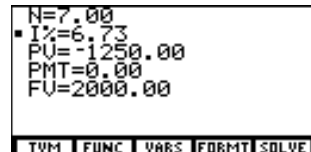
- Display the TVM Solver. Enter 7 annual payments, present value of -\$1,250 (negation indicates cash outflow), payment amount of \$0, and future value of \$2,000.

[F1] 7  $\downarrow$   $\downarrow$  (-) 1250  
 $\downarrow$  0  $\downarrow$  2000 [ENTER]



- Move the cursor to **I% =** (interest rate) and then select **SOLVE** from the TVM Solver menu. A small square is displayed next to the solution. The solution value is stored to the TVM variable **I**.

$\uparrow \uparrow \uparrow$  [F5]



## Using the Financial Functions

### Entering Cash Inflows and Cash Outflows

When using the financial functions, you must enter cash inflows (cash received) as positive numbers and cash outflows (cash paid) as negative numbers. The financial functions follow this convention when computing and displaying answers.

### FIN FUNC (Financial Functions) Menu [2nd] [MATH] [MORE] [F2]

TVM	FUNC	VARS	FORMT	Uninst						
tvmN	tvmI	tvmPV	tvmP	tvmFV	▶	npv	irr	bal	Σprn	Σint
					▶	nom	eff	dbd	pBegin	pEnd

### Calculating Time-Value-of-Money

The first five items on the FIN FUNC menu are the time-value-of-money (TVM) functions. You can use them to analyze financial instruments, such as annuities, loans, mortgages, leases, and savings, on the home screen or in a program.

*All arguments and punctuation inside the [ ] brackets are optional.*

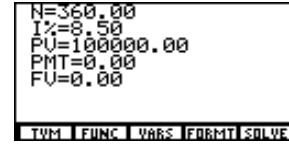
- tvmN** [(I%,PV,PMT,FV,P/Y,C/Y)]      Computes the number of payment periods
- tvmI** [(N,PV,PMT,FV,P/Y,C/Y)]      Computes the annual interest rate
- tvmPV** [(N,I%,PMT,FV,P/Y,C/Y)]      Computes the present value
- tvmP** [(N,I%,PV,FV,P/Y,C/Y)]      Computes the amount of each payment
- tvmFV** [(N,I%,PV,PMT,P/Y,C/Y)]      Computes the future value

*To store a value to a TVM variable, use the TVM Solver or use [STO▶] and any TVM variable on the FIN VARS menu.*

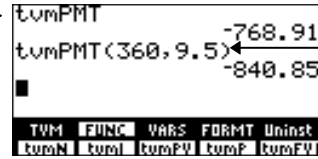
Each TVM function takes zero to six arguments. Each argument must be a real number or a TVM variable. The values that you specify as arguments for these functions are not stored to the TVM variables.

If you enter less than six arguments, you must enter arguments in the order of the syntax, up to the last argument you want to enter. The program substitutes a previously stored TVM variable value for each subsequent unspecified argument. If you enter any arguments with a TVM function, you must place the argument or arguments in parentheses. The following examples show some ways to use the TVM functions.

Assume these values are stored to the TVM variables in the payment format editor and TVM Solver.

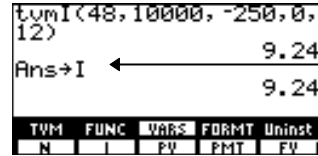


When you execute a TVM function on the home screen with no specified arguments, the TVM function (**tvmPMT** in the example) uses stored TVM variable values.



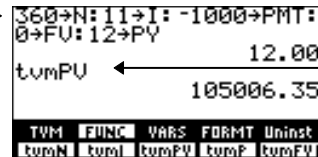
To change an argument without changing the value stored to a TVM variable, enter arguments up to the argument you want to change. In the example, the interest rate is changed to 9.5.

You can enter arguments directly on the home screen. Remember, neither the answer nor the arguments are stored to the TVM variables.



To store an answer to the appropriate TVM variable, use **[STO]** and the FIN VARS menu.

If you prefer, you can store values to the TVM variables on the home screen.



When you execute a TVM function (**tvmPV** in the example), it uses the newly stored TVM variable values.

### Calculating Cash Flows

The next FIN FUNC menu items are cash flow functions. Use them to analyze the value of money over equal time periods. You can enter unequal cash flows. You can enter cash inflows or outflows.

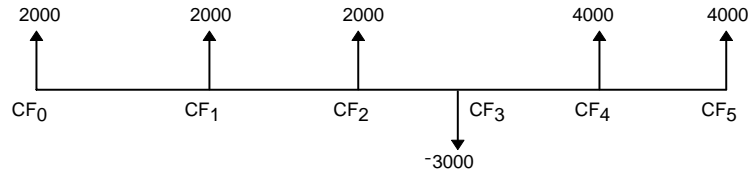
**npv**(*interestRate*, *cashFlow0*, *cashFlowList*[, *cashFlowFrequency*]) Returns the sum of the present values for the cash inflows and outflows

**irr**(*cashFlow0*, *cashFlowList* [, *cashFlowFrequency*]) Returns the interest rate at which the net present value of the cash flows is equal to 0

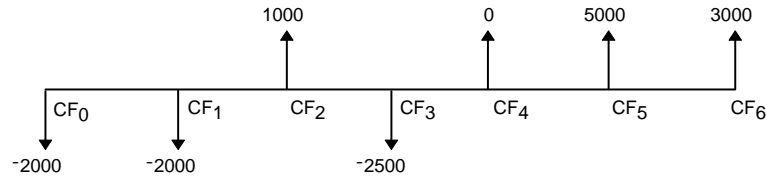
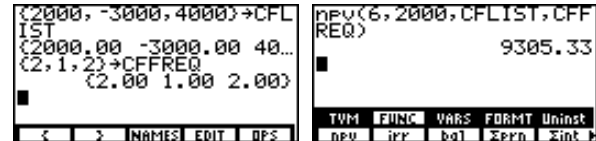
- ◆ *interestRate* is the rate by which to discount the cash flows (the cost of money) over one period.
- ◆ *cashFlow0* is the initial cash flow at time 0; it must be a real number.
- ◆ *cashFlowList* is a list of cash flow amounts after the initial cash flow *cashFlow0*.
- ◆ *cashFlowFrequency* is a list in which each element specifies the frequency of occurrence for a grouped (consecutive) cash flow amount, which is the corresponding element of *cashFlowList*. The default is 1; if you enter values, they must be positive integers <10,000.



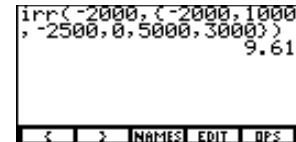
The uneven cash flow below is expressed in lists. *cashFlowFrequency* indicates that the first element in *cashFlowList* (2000) occurs twice (2), the second element (-3000) occurs once (1), and the third element (4000) occurs twice (2).



*cashFlow0* = 2000  
*cashFlowList* = {2000, -3000, 4000}  
*cashFlowFrequency* = {2,1,2}  
*I%* = 6



*cashFlow0* = -2000  
*cashFlowList* = {-2000, 1000, -2500, 0, 5000, 3000}  
*cashFlowFrequency* = N/A



### Calculating Amortization

Items eight, nine, and ten are the amortization functions. Use them to calculate balance, sum of principal, and sum of interest for an amortization schedule.

*roundValue* specifies the internal precision used to calculate the balance.

*effectiveRate*, *nominalRate*, and *compoundingPeriods* must be real numbers; *compoundingPeriods* must be > 0.

*bal*( , *ΣPrn*( , and *ΣInt*( use stored values for *I%*, *PV*, and *PMT*. You must store values to these variables before computing the principal.

**bal**(*nPayment*[,*roundValue*])

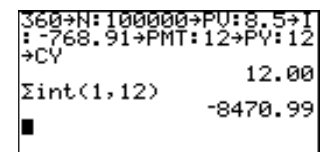
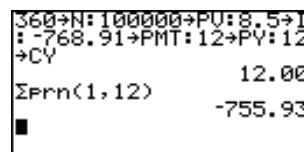
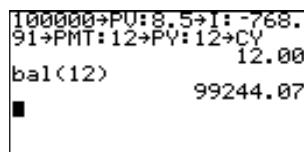
Computes the balance for an amortization schedule; *nPayment* (the number of the payment at which to calculate a balance) must be a positive integer <10,000

**ΣPrn**(*paymentA*,*paymentB* [,*roundValue*])

Computes the sum of the principal during a specified period for an amortization schedule; *paymentA* (the starting payment) and *paymentB* (the ending payment in the range) must be positive integers <10,000



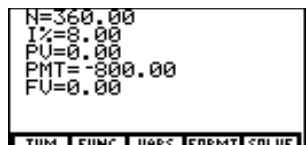
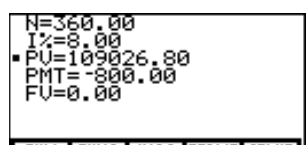

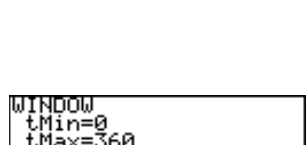
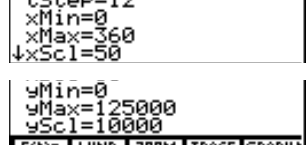
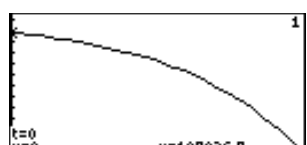

**ΣInt**(*paymentA*,*paymentB* [,*roundValue*])

Computes the sum of the interest during a specified period for an amortization schedule; *paymentA* (the starting payment) and *paymentB* (the ending payment in the range) must be positive integers <10,000



### Amortization Example: Calculating an Outstanding Loan Balance

You want to buy a home with a 30-year mortgage at 8 percent annual percentage rate (APR). Monthly payments are \$800. Calculate the outstanding loan balance after each payment and display the results in a graph and in the table.

- |   |  |   |
|---|--|---|
| <p>1 Display the mode screen and set the fixed-decimal setting to 2, as in dollars and cents. Also, set Param graphing mode.</p>                            | <p><b>2nd</b> <b>[MODE]</b> <b>↓</b><br/> <b>▶▶▶</b> <b>[ENTER]</b><br/> <b>↓↓</b> <b>▶▶</b><br/> <b>[ENTER]</b></p>   |    |
| <p>2 Display the payment format editor, and then set payments and compounding periods per year to 12, to be received at the end of each period.</p>         | <p><b>2nd</b> <b>[MATH]</b> <b>[MORE]</b> <b>[F2]</b><br/> <b>[F4]</b> <b>12</b> <b>↓</b> <b>[ENTER]</b></p>   |    |
| <p>3 Display the TVM Solver, and then enter the known TVM variable values:<br/> <b>N=360</b>      <b>PMT= -800</b><br/> <b>I=8</b>          <b>FV=0</b></p> | <p><b>[F1]</b> <b>360</b> <b>↓</b> <b>8</b> <b>↓</b><br/> <b>(-)</b> <b>800</b> <b>↓</b> <b>0</b> <b>[ENTER]</b></p>   |    |
| <p>4 Move the cursor to the <b>PV=</b> prompt and solve for the present value of the loan. A small square specifies the solution.</p>                       | <p><b>↑↑</b> <b>[F5]</b></p>   |    |
| <p>5 Display the parametric equation editor. Turn off all stat plots.</p>   | <p><b>[GRAPH]</b> <b>[F1]</b> (if a plot is on, press <b>↑</b>, <b>↓</b> to highlight it, and <b>[ENTER]</b>); then <b>↓</b>)</p>  |   |
| <p>6 Define <b>xt1</b> as <b>t</b> and <b>yt1</b> as <b>bal(t)</b>.</p>   | <p><b>[F1]</b> <b>↓</b> <b>2nd</b> <b>[MATH]</b><br/> <b>[MORE]</b> <b>[F2]</b> <b>[F2]</b> <b>[MORE]</b><br/> <b>[F3]</b> <b>2nd</b> <b>[F1]</b> <b>↓</b></p>   |  |
| <p>7 Display the window variable editor, and then enter these window variable values as shown.</p>  | <p><b>[GRAPH]</b> <b>[F2]</b> <b>0</b> <b>↓</b> <b>360</b><br/> <b>↓</b> <b>12</b> <b>↓</b> <b>0</b> <b>↓</b> <b>360</b><br/> <b>↓</b> <b>50</b> <b>↓</b> <b>0</b> <b>↓</b><br/> <b>125000</b> <b>↓</b> <b>10000</b></p> |  |
| <p>8 Draw the graph and activate the trace cursor. Explore the graph of the outstanding balance over time.</p>  | <p><b>[F5]</b><br/> <b>[F4]</b><br/> <b>←</b> <b>→</b></p>   |  |
| <p>9 Enter a value for <b>t</b> to view the balance at a specific time.</p>   | <p><b>24</b> <b>[ENTER]</b></p>  |  |

A stat plot is turned on if it is highlighted with a box.

- ⑩ Display the table setup editor, and then enter these values:  
**TblStart=0**    $\Delta$ **Tbl=12**  
**Indpnt: Auto**

TABLE [F2]  
 0 [v] 12 [v] [ENTER]

```
TABLE SETUP
TblStart=0
ΔTbl=12
Indpnt: AUTO Ask
```

- ⑪ Display the table of outstanding balances, where **xt1** represents time and **yt1** represents balance at that point in time.

[F1]

t	xt1	yt1
0.00	0.00	109026.8
12.00	12.00	108116.0
24.00	24.00	107129.7
36.00	36.00	106061.5
48.00	48.00	104904.6
60.00	60.00	103651.6

t=0

TBLSET	SELECT	t	xt	yt
--------	--------	---	----	----

### Calculating Interest Conversion

Use the interest conversion functions **nom** and **eff** to convert interest rates from an annual effective rate to a nominal rate (**nom**), or from a nominal rate to an annual effective rate (**eff**).

**nom**(effectiveRate,compoundingPeriods)    Computes the nominal interest rate

**eff**(nominalRate,compoundingPeriods)    Computes the effective interest rate

```
nom(15.87,4)    15.00
```

```
eff(8,12)    8.30
```

### Finding Days Between Dates

Use the date function **dbd** to calculate the number of days between two dates using the actual-day-count method. *dateA* and *dateB* can be numbers or lists of numbers within the range of the dates on the standard calendar.

**dbd**(dateA,dateB)    Calculates the number of days between dates; enter *dateA* and *dateB* in either of two formats: *MM.DDYY* (for U.S.) or *DDMM.YY* (for Europe)

```
dbd(12.3190,12.3192)
731.00
```

TVM	FUNC	VARS	FORMAT	Uninst
nom	eff	dbd	FRS3in	pEnd

*Dates must fall between the years 1950 and 2049.*

### Defining the Payment Method

**pEnd** and **pBegin** specify a transaction as an ordinary annuity or an annuity due. Executing either instruction sets the payment method for subsequent financial calculations. The current setting is displayed in the payment format editor (page 4).

**pBegin**    Specifies an annuity due, where payments occur at the beginning of each payment period (Most leases are in this category.)

**pEnd**    Specifies an ordinary annuity, where payments occur at the end of each payment period (Most loans are in this category; **Pmt\_End** is the default.)

*On the payment format editor's **PMT:END BEGIN** line, select **END** to set ordinary annuity or select **BEGIN** to set annuity due.*

## Menu Map for Financial Functions

MATH Menu (where FIN is automatically placed)  $\boxed{2nd}$   $\boxed{[MATH]}$

NUM	PROB	ANGLE	HYP	MISC	▶	INTER	FIN			
-----	------	-------	-----	------	---	-------	-----	--	--	--

(MATH) FIN (Financial) Menu  $\boxed{2nd}$   $\boxed{[MATH]}$   $\boxed{MORE}$   $\boxed{F2}$

TVM	FUNC	VARS	FORMT	Uninst
-----	------	------	-------	--------

FIN TVM (Time-Value-of-Money) Solver Menu  $\boxed{2nd}$   $\boxed{[MATH]}$   $\boxed{MORE}$   $\boxed{F2}$   $\boxed{F1}$

TVM	FUNC	VARS	FORMT	SOLVE
-----	------	------	-------	-------

FIN FUNC (Financial Functions) Menu  $\boxed{2nd}$   $\boxed{[MATH]}$   $\boxed{MORE}$   $\boxed{F2}$   $\boxed{F2}$

TVM	FUNC	VARS	FORMT	Uninst	▶	npv	irr	bal	$\Sigma$ prn	$\Sigma$ int
tvmN	tvmI	tvmPV	tvmP	tvmFV	▶	nom	eff	dbd	pBegin	pEnd

FIN VARS (Financial Variables) Menu  $\boxed{2nd}$   $\boxed{[MATH]}$   $\boxed{MORE}$   $\boxed{F2}$   $\boxed{F3}$

TVM	FUNC	VARS	FORMT	Uninst	▶	PY	CY			
N	I	PV	PMT	FV	▶					

FIN FORMT (Financial Format) Menu  $\boxed{2nd}$   $\boxed{[MATH]}$   $\boxed{MORE}$   $\boxed{F2}$   $\boxed{F4}$

TVM	FUNC	VARS	FORMT	
-----	------	------	-------	--