



Programmable Calculator News

Volume 1, Number 1



Letter From The Editor

Texas Instruments welcomes you to the first edition of Programmable Calculator News. We're excited about advancements like "Menu Windows", redefinable function keys, and BASIC problem-solving power that make this a new era in calculator technology.

Our publication will regularly feature program listings, technical articles, question and answer segments, new product reviews, and user comments. Information on new products will be included in future issues. Though articles in this edition have been contributed by members of our technical staff, we invite your knowledge and experience. Simply address articles, questions, or comments to:

The Editor
Programmable Calculator News
P.O. Box 53
Lubbock, TX 79408



Free Newsletter

Would you like us to send the newsletter to a friend?

Name

Home Address

City

State Zip

Industry

Job Function

Mail to:
Programmable Calculator News
P.O. Box 53
Lubbock, TX 79408

Programmable Calculator News is a periodic publication.
Copyright 1987 by Texas Instruments, Inc.



TI-95 PROCALC™

A New Standard of Comparison

The thrust of TI-95 PROCALC™ design was to provide as much computing power as possible while improving ease of use by introducing "menu windows" to programmable calculators. Menus of functions are used to avoid difficult-to-remember key sequences and symbol-cluttered keyboards. Extensive prompting in the alphanumeric display guides the user through otherwise complex functions. Messages are also used to provide an explanation when errors occur. These techniques allow state-of-the-art capability to be presented to the user in an easy-to-learn, easy-to-use manner.

IC Technology

To minimize system complexity, the TI-59 used a custom set of integrated circuits that was designed specifically for calculator computations. Custom calculator chips are still required for complete integration on single chip calculators. However, the increased amounts of memory in the TI-95 made system flexibility and standard memory interface more important design considerations. Because of these factors, the TI-95 uses a TMS-70C46

microprocessor which is a custom version of TI's standard TMS7000 family. Smaller IC geometries were the key to the development of improved architectures, like the TMS70C46, which are more flexible than the TI-59 chip set and also allowed production of ROMs and RAMs with much larger capacities. The switch from PMOS calculator chips in 1977 to CMOS in newer designs has improved battery life enormously in new products.

RAM Cartridges Replace Magnetic Cards

One of the most difficult aspects of the TI-59 design was to provide a convenient, low-cost method of saving programs and data. The TI-59 solution was a magnetic card reader/writer which could save 480 program steps per card. Magnetic cards were inexpensive and small enough for users to carry their entire library of programs in a pocket. Although this provided a workable solution, the magnetic card mechanism required very careful alignment and testing procedures and was, therefore, difficult to manufacture.

Because of the availability of large static RAMs, the TI-95 is able

In This Issue:

	Page
TI-95 PROCALC™: A New Standard of Comparison	TI-95 2-3
Introducing the Concept of Menu Functions	TI-95 4
Indirect Addressing	TI-95 4-5
TI-74 BASICALC™	TI-74 5-6
Using the ACCEPT Statement	TI-74 6-7
Trig in BASIC	TI-74 7
Can I Really Afford It?	TI-74 7-8

to offer convenient storage without the need for a magnetic card mechanism. Instead, it has a system for saving programs and data as named files in RAM. Up to 6200 bytes or program steps can be saved in the built-in RAM, and additional capacity is provided by Constant Memory™ Cartridges which offer convenient, removable file storage. These cartridges contain an 8K byte RAM and a lithium battery to maintain data for up to five years when the cartridges are unplugged from the calculator.

The number of programs which can be saved varies with program size. For example, one Constant Memory cartridge can save 8 programs with over 1000 steps each or 40 programs with about 200 program steps each. Since keystroke programs tend to be compact, each cartridge can save many useful programs and associated data. The CI-7™, a cassette interface cable, is also available to save programs and data on an audio cassette recorder.

Programs can be run directly from either the built-in file space or a Constant Memory cartridge. When the Run key is pressed, the user is given the choice of running programs from any of the available memory areas. After choosing the cartridge or built-in file area, program names appear in the menu windows and the user pushes one of the function keys to run the desired program. This makes all stored programs easy to access and eliminates the need to download before execution.

TI-95 PROCALC Features

In addition to providing the scientific functions of the TI-59, the TI-95 adds factorial, hyperbolic trigs, metric conversions, permutations and combinations, roots of cubic and quadratic equations, random-number generation, prime factors, greatest common divisor, and least common multiple. It also adds the capability to convert between number bases and perform calculations in decimal, octal, or hexadecimal.

The TI-95 has extensive alphanumeric capability and easy-to-use keystroke programming functions that allow the user to create his own messages to be displayed. The alpha keys are also used to enter program labels, register names, and file names. To support the use of

	TI-59	1977	TI-95	1986
MEMORY:				
ROM	6K x 13 bits		36K x 8 bits	
ROM Cartridges	5K bytes		32K bytes	
RAM	1K bytes		8K bytes	
Maximum Program Size	960 steps		7200 steps	
Maximum Registers	100		900	
Program Storage	Magnetic cards: 480 steps/card		Built-in files: 6200 steps maximum RAM cartridges: 8K steps/cartridge	
DISPLAY:				
Type	7 segment LED		5 x 7 Dot-matrix LCD	
Number of characters	12		31	
Special indicators	0		17	
KEYBOARD:				
Number of keys	45		65	
Arrangement	Vertical		Horizontal: QWERTY	
FUNCTIONALITY:				
Number of functions	Over 100		Over 200	
Number of program codes	100		251	
Printable characters	64		95 ASCII	
SPEED:				
Bond program benchmark	38.5 sec		8.0 sec	
BATTERY LIFE:				
Continuous Execution	5 hours		Over 100 hours	
INTEGRATED CIRCUITS:				
	9 PMOS 4 bit		5 CMOS 8 bit	
	1 TMC0501 CPU		1 TMS70C46 CPU	
	4 240 byte RAM		1 8K byte RAM	
	2 2.5K word ROM		1 32K byte ROM	
	1 1K word ROM			
	1 Mag Card Chip		2 Display Drivers	

alphanumeric features, the TI-95 has a large, positive-response keyboard with the alphabet arranged in a typewriter (QWERTY) format.

Expansion

Application programs are available on 32K byte ROM cartridges—more than six times the size of the application cartridges that were available for the TI-59. Math, Statistics, and Chemical Engineering are the first to be offered, but cartridges covering other subjects are being developed. RAM cartridges are used for saving programs and data. In addition, a general purpose input/output system is built into the TI-95 and an I/O port on the back of the unit is used for attaching peripherals. The peripherals presently available are the CI-7 audio cassette interface and the PC-324™, a 24-column, portable, thermal printer.

Operation Speed

The TI-95 PROCALC executes similar calculator functions much faster than the TI-59. The benchmark program referred to in Figure 1 calculates the present value of a bond. It is mostly arithmetic

execution and the TI-95 completes the evaluation in about one fifth the time taken by the TI-59. In operations that involve mostly data transfers and manipulation, the TI-95 operates about ten times the TI-59 speed.

Summary

The technology of advanced calculators has changed significantly in recent years. The features that were a triumph for the TI-59 are now easily accomplished. Although the TI-59 was considered the standard for comparison in 1977, the TI-95 PROCALC has more memory, more speed, and about twice as many functions. The challenge today is to present this formidable power to the user in a manner which makes it easy to understand and use. The TI-95 uses an alphanumeric display and menus of functions to accomplish this and, in the process, sets new standards by which advanced scientific programmables will be judged. ■

TI-74 BASICALC, TI-95 PROCALC, Constant Memory, CI-7, and P-324 are trademarks of Texas Instruments.

The previous information, excerpted from an article by Arthur C. Hunter, is reprinted with permission from the July-August Edition of *TJ Technical Journal*.

Introducing the Concept of Menu Functions

by Ash Osmani

The TI-95 PROCALC™ is the first programmable calculator to introduce the concept of "menu windows". Redefinable function keys (F1-F5) select function definitions from five menu windows located below the display line. This allows both experienced and inexperienced users quick access to any desired program location.

System Menu Functions

The system functions have been organized by menus. To select a function, depress one of 9 menu keys, such as "CONV", then use F1-F5 to choose the desired function. Functions can be accessed quickly since menu organization allows a maximum of 2-levels in selection.

Creating Custom Menus

Display Windows and Function Keys can be used to create custom menu functions. Menu windows allow the user to select any desired program from user memory, built-in file, or application cartridge. Stored programs are easy to access and the need to download before execution is eliminated.

Table 1 shows an example program that creates a menu to determine the nature of the number in display, odd or even (O/E), the number of digits in display (DIG), and the sum of digits in display (SUM). The program has been designed so that each function key (F1-F3) transfers control to a specific program segment which computes one of the above functions. After computations, a return (RTN) is used to transfer control back to the main menu. The menu window corresponding to F5 has been used for escape (ESC) to clear the display and menu windows.

Figure 1 illustrates our created menu. Two keystrokes, RUN and PGM, activate the program. The menu windows have been defined using the DFN function (see Table 1). The DFN function is used to label keys F1, F2, F3, & F5 using display windows and defines the program labels AA, AD, AG, and AJ. The DFN function can be used

repeatedly in any program to update definitions of the functions as choices are selected. User-defined menus of custom functions can, therefore, be created even in problems requiring multiple menus.

The number 123 has been used as an input (Figure 1B). Output display of computations is shown in Figure 1 (C, D, E, & F). Please note: the program accepts only positive integers. Any other will result in the error message "INVALID NUMBER".

```

0000 LBL AC CE 'EXAMPLE'
0011 ' PROGRAM' DFN CLR
0021 DFN F1:O/E@AA
0028 DFN F2:DIG@AD
0035 DFN F3:SUM@AG
0042 DFN F5:ESC@AJ RTN
0050 LBL AA SBL AK TF 00
0058 RTN (( STO B /2)
0066 INT *2) IF= B
0072 GTL AB CE 'ODD NUM'
0083 'BER' RTN
0087 LBL AB CE 'EVEN NU'
0098 'MBER' RTN
0103 LBL AD SBL AK TF 00
0111 RTN x-t 0 STO B 1
0117 STO X x-t
0120 LBL AE (/10)INT
0129 IF= B GTL AF INC X
0136 GTL AE
0139 LBL AF RCL X COL 06
0146 MRG X COL 06 ' DIG'
0154 'ITS' RTN
0158 LBL AG SBL AK TF 00
0166 RTN x-t 0 STO B
0171 STO C x-t
0174 LBL AH (/10)STO A
0184 ( FRC *10) ST+ C
0192 RCL A INT IF= B
0197 GTL AI GTL AH
    
```

```

0203 LBL AI CE ' SUM='
0212 COL 14 MRG C COL 14
0218 RTN
0219 LBL AJ DFN CLR 0
0225 RTN
0226 LBL AK RF 00 STO B
0233 ABS INT INV IF= B
0238 GTL AL 0 IF= B
0244 GTL AL RCL B RTN
0250 LBL AL 'INVALID NU'
0263 'MBER' SF 00 RTN
    
```

Press RUN, PGM



Enter 123



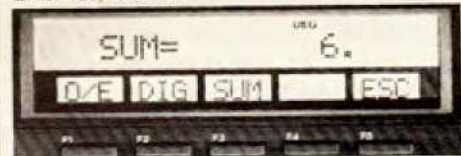
Press F1



Enter 123, Press F2



Enter 123, Press F3



Press F5



Indirect Addressing

By Rena Gillis

Indirect addressing can be very useful when you are going to have a number of entries and wish to store each entry in the calculator for future calculation. In the following example, indirect addressing is used to store and then subsequently recall as many entries as the unit is partitioned to accept. If addressed directly, the program would become

uncommonly bulky and complicated, making it difficult to store as well as run.

Indirect addressing uses one memory register as a "pointer" which directs any subsequent data to be stored or recalled to the memory address contained in this register. This number can be incremented or decremented by the program. In this

program, a number 11 is stored in our "pointer" register (I) at the beginning. At LBL CC we see this same register being incremented by 1, bringing its total to 12. The amount paid for this first transaction is then stored in register 12. We see register I being incremented again by 1. The grain weight is recalled and then stored in register 13 as directed by our "pointer" register. Later in the program, at LBL TA, indirect addressing is used to print out a tally of the daily grain receipts and cash outflows.

AAA Grain Mills, Inc. receives several truck loads of grain per day from area farmers during the harvest season. After weighing the full vehicles, the load is emptied, the grain evaluated for grade, and the vehicle reweighed. The following program calculates the weight of the grain and amount due to the carrier then stores the data for later retrieval when it gives the day's total receipts.

This program allows a daily change of price per cwt. (hundred-weight) for two different grades. At the end of the day, the operator needs only to hit "TOT" and the calculator will give a grand total of all grain received and monies paid out with a listing of each individual transaction. Each time the program is run, all memories are cleared, therefore, no data from the previous run will be retained. If a full weight is less than an empty weight, an error message will be displayed and the calculator will ask you to re-enter that particular transaction.

Example:

Grade A is selling at \$2.87/cwt.
Grade B is selling at \$2.65/cwt.

Full Wt.	Empty Wt.	Grade
5,283	3,532	A
11,263	6,453	A
12,422	6,250	B
10,890	5,750	A

```

0000 'DLY GRAIN REC'TS'
0016 PAU FIX 2 CMS 11
0022 STO I 'ENT GRADE A'
0035 '$' BRK /100=STO A
0044 'ENT GRADE B$' BRK
0057 /100=STO B
0064 LBL MI CLR 'ENT FU'
0074 'LL WT'
0079 DFN F1:ENT@CL
0086 DFN F5:TOT@TL HLT
0094 LBL ER 'ERROR' PAU
0103 CLR 'RE-ENTER FIGU'

```

```

0117 'RES' PAU CLR
0122 GTL MI
0125 LBL CL PRT STO C
0131 CLR 'ENT EMPTY WT'
0144 BRK PRT STO D CLR
0149 RCL C - RCL D =
0155 STO E PRT IF< K
0160 GTL ER ST+ F CLR
0166 'GRADE?' DFN CLR
0174 DFN F1: A @CA
0181 DFN F5: B @CB HLT
0189 LBL CA RCL E *
0195 RCL A = GTL CC
0201 LBL CB RCL E *
0207 RCL B =
0210 LBL CC STO G ST+ H
0217 PRT INC I STO IND I

```

```

0223 INC I RCL E
0227 STO IND I ADV
0231 GTL MI
0234 LBL TL ADV CLR '# '
0241 'GRAIN' PRT RCL F
0249 PRT ADV '$ PAID OU'
0260 'T' PRT RCL H PRT
0265 ADV 1 STO J 12
0271 STO I
0273 LBL TA RCL J PRT
0279 RCL IND I IF= K
0284 GTL TB PRT INC I
0290 RCL IND I PRT ADV
0295 INC I INC J GTL TA
0302 LBL TB 'END' PRT
0309 ADV HLT

```



TI-74 BASICALC™

The TI-74 BASICALC™, designed for students, engineers, and scientists, combines advanced BASIC language programmability and scientific calculator functions in a battery operated portable unit with system expansion capability.

BASIC Programmability

The BASIC mode provides access to a full featured, high level programming language. BASIC on the TI-74 has been implemented in close compliance with ANSI minimal standards for BASIC language com-

puters. This means that the TI-74's BASIC language is similar to the BASIC language implementations on larger computers rather than the reduced capabilities which are usually associated with calculator-type BASIC machines. For example, many BASIC programs written for the TI 990 mini-computer will run on the TI-74 with few or no modifications.

In addition to being designed with ANSI standards in mind, many enhancements have also been specifically included for use on a portable calculator. Numeric or

BASIC expressions entered without line numbers are executed in an immediate mode, whereas expressions entered with line numbers become entries into the BASIC program. This ability to switch quickly from program entry to immediate execution makes the product more versatile.

The TI-74 has 8K Constant Memory™ RAM for medium sized applications. The unit's storage capacity can be doubled with the addition of an 8K Constant Memory™ cartridge. Programs or data can be stored off-line with the CI-7 Audio Cassette Interface. The TI-74 system includes the optional PC-324, a 24 column thermal printer for applications requiring hard copy of program listings or results.

Scientific Calculator

In the calculator mode, the TI-74 offers 70 scientific functions, alphanumeric error messages, and 13-digit accuracy which spans a numeric range of $\pm 9.999999999999999E127$. Features include logarithmic, hyperbolic, and trigonometric functions, two variable statistics, and probability calculations. Ten user registers are available for saving intermediate results or constants.

Ease of Use

The TI-74 system includes several features to facilitate operation and use. The large 31 digit alphanumeric LCD is easy to read and software scrolling with cursor keys supports a maximum of 80 characters per line. Fourteen annunciators indicate errors, modes of operation, and machine status. INSERT and DELETE keys assist in source program editing. The keyboard consists of 62 keys arranged in a QWERTY (typewriter) layout with large format keys and a numeric keypad. Common BASIC language keywords are defined on the keyboard and can be entered with only two keystrokes. The system includes a durable carrying case which also holds a quick reference card. The documentation that is provided consists of two books, a User's Guide and a Programming Guide.

Expansion

The TI-74 console contains a cartridge port to augment standard memory. An 8K RAM cartridge can

be software configured to double the amount of memory available for programs/data or configured as an 8K mass storage device. Another option is the use of 32K ROM cartridges which contain application programs designed to solve specific problems. Mathematics, Statistics, Chemical Engineering, Finance, and Learn Pascal ROM cartridges are currently available.

The console interfaces to a battery operated thermal printer and audio cassette interface via an I/O port. When connected to the printer, the

TI-74 and cassette interface are powered by the printer's batteries or AC adapter, allowing the entire system to operate from one power source. These accessories round out a complete system designed to introduce the beginning user to the world of programming and solve complex problems for engineers and other advanced users. ■

The previous article by Bob Hanschke is reprinted with permission from the March/April 1986 *TI Engineering Journal*.

Using the ACCEPT Statement

by Stephen L. Reid

The ACCEPT statement is a very versatile feature of TI-74 BASIC. With its many options, ACCEPT can provide keyboard data entry filtering for nearly any application. The main purpose of this article is to demonstrate use of the ACCEPT statement when several inputs are desired with all prompts displayed at the same time. This type of application occurs when prompts are short, the information to be read is related, and all information is to be displayed at once.

For example, let's request that the user enter height, weight, and age. We want all prompts to be visible at the same time. The user can move from one field to the next while data from the previous response remains displayed. Prompts for this example can be entered by using PRINT or DISPLAY statements. While multiple statements may be needed, remember that these PRINT and DISPLAY operations should be PENDING PRINT or DISPLAYs. This is done by placing a semi-colon (;) at the end of the print list associated with PRINT or DISPLAY. If prompts are not left in a pending condition, ACCEPT clears the display prior to waiting for data.

For our height, weight, and age example, the PRINT statement to place a prompt in the 31 column display would be:

```
100 Print "Ht:5' 11"  
Wt:155lbs Age: yrs";
```

This prompt has 4 fields. One field is in column 4 and is one character wide (the Feet Field). The next field begins in column 7 and is 2 characters wide (the Inches Field). Next is the Weight Field which begins in column 15 and consists of three characters. The last field is the Age Field, two characters wide, beginning in column 27. To ACCEPT data from these fields, the following ACCEPT statements are used:

```
110 ACCEPT AT(4) SIZE(-1)  
VALIDATE (DIGIT),FEET$
```

```
120 ACCEPT AT(7) SIZE(-2)  
VALIDATE(DIGIT),INCH$
```

```
130 ACCEPT AT(15) SIZE(-3)  
VALIDATE(DIGIT),WT$
```

```
140 ACCEPT AT (27) SIZE(2)  
VALIDATE(DIGIT)  
NULL("00"),AGES
```

In each ACCEPT Statement, the AT() option specifies the beginning column of the field. The SIZE() option specifies the width, and the VALIDATE() option specifies DIGIT, limiting accepted keys to digits 0 through 9. Other key presses (except for BREAK) will be ignored by ACCEPT. A minus sign in the SIZE() option tells ACCEPT not to clear the input field prior to accepting data. This allows default values to be displayed with the prompt. If SIZE is used with a non-negative

field width, the field is cleared before data entry begins.

The ACCEPT Statement in line 140 has a NULL() option. The value of the NULL parameter will be assigned to the variable used with ACCEPT if the user presses ENTER when ACCEPT is on an empty field. The Age Field was left blank by the prompt, so if the user presses ENTER in response to this prompt, the string value "00" will be assigned to the variable AGES.

Minor changes in options of the ACCEPT statement will change the way in which it works. For example, if we want to limit the acceptable range in the Feet Field to 4 through 7, then changing the VALIDATE() option to VALIDATE("4567") will accomplish this. The user could no longer enter 0 to 3, 8, or 9 as input to the Feet Field. This single feature can save a lot of additional testing with IF statements after inputting data, since the data has already been verified within the program range. Multiple test fields can be given to the VALIDATE option separated by commas. If only digits and certain characters, such as "Y", "N", and spaces are wanted, the VALIDATE() option could become VALIDATE(DIGIT, "YN").

Combined with clear and precise prompts, the ACCEPT statement and its options provide a powerful and flexible keyboard data input tool for the TI-74. It can handle all data entry and validation that most programs require while reducing program size by eliminating the need for input testing code. ■

Trig in BASIC?

by Patrick Hicks

When I discovered there was to be a user newsletter for the TI-74, I was elated. I have been exposed to this machine for only a short time, but have become addicted to its unique features. How could those guys squeeze so much into such a small package? The manual just can't begin to cover everything the TI-74 can do.

One of the first things I noticed was the CONVERSION FEATURE in Calculator Mode. One of its uses is a conversion from Degrees-Minutes-Seconds into Decimal Degrees—a pretty important feature

since all Trig calculations are interpreted as Decimal Degrees. Anyone in Engineering, Surveying, or Navigation will readily agree that such a feature is a must.

Much to my disappointment, I was unable to find a comparable conversion in the BASIC Mode. Had the design department brains simply overlooked this one? Apparently not. When I researched my query further, I could not find the conversion in any BASICs I checked. Therefore, I decided to write a subprogram to do the "dirty work" for me and here it is.

```
100 INPUT "ANGLE(DMS)=";A
110 CALL CONVERT(A)
120 PRINT "CONV. ANGLE(DD)=";
A:PAUSE
300 SUB CONVERT(X)
310 IF X=INT(X)THEN 400
320 TEMPS=STR$(X)
330 FOR I=1 TO LEN(TEMPS)
340 IF SEGS(TEMPS,I,1)="."
THEN 355
350 NEXT I
```

```
355 IF I=1 THEN X=0:GOTO 370
360 X=VAL(SEGS(TEMPS,I,1))
370 X=X+(VAL(SEGS(TEMPS,I,
3))/.6)
380 TEMPS="."00&SEGS(TEM
PS,I+3,LEN(TEMPS))
390 X=X+(VAL(TEMPS)/.36)
400 SUBEND
```

In line 100, the program will prompt you to enter any angle in Degrees-Minutes-Seconds Format (DD.MMSS) and press ENTER. Next, line 110 calls the subprogram CONVERT (lines 300-400) then passes the entered variable (A) to the routine. Finally, the subprogram returns (A) back to the calling program to display the angle in its converted form. To use this simple subprogram in your applications, simply add the contents of lines 300-400 to the END of your program and use the CALL CONVERT(numeric variable) whenever you need to convert an angle from Degrees-Minutes-Seconds to Decimal Degrees. ■

Can I Really Afford It?

by Patrick Hicks

Since interest rates available on new automobiles are at record lows, I've been out there like most of you looking at new cars. If you are like me, you can't afford to pay cash for a twelve-thousand-dollar car. You do know about how much you can afford for a monthly payment, but one question arises, "How does the payment I can afford relate to dealer sticker price?"

I realized that my faithful TI-74 could help on this one, and after a bit of research I found the following equations. The first determines the sticker price to find when I know my affordable payment amount, the annual interest rate (APR), and the total number of payment periods in the contract.

$$PV = PMT \left(\frac{1 - (1+i)^{-n}}{i} \right)$$

The next one shows payment amount if I know the annual interest rate (APR), the purchase

amount of the auto, and total number of payment periods.

$$PMT = PV \left(\frac{i}{1 - (1+i)^{-n}} \right)$$

The following program incorporates these formulas into TI-74 BASIC. Lines 100-150 are simply a menu that outlines your options. Lines 160-210 are used to calculate the amount you can afford to finance (or PV). Lines 220-250 calculate the payment. Finally, line 260 ends the program and returns you to the BASIC Immediate Mode. The program is simple so it does not account for zero percent financing, down payments, or any fees associated with your purchase. It is meant to give ideas on what you can or can not afford.

```
100 PRINT "PRESS:" :PAUSE 1
110 PRINT "1) TO COMPUTE PV":
PAUSE 2
120 PRINT "2) TO COMPUTE PAYMENTS":
PAUSE 2
```

```

130 PRINT "3) TO SEE OPTIONS
AGAIN":PAUSE 2
140 PRINT "4) END":PAUSE 2
150 INPUT X:ON X GOTO 160,160,
100,260
160 INPUT "INTEREST=";I:I=I/1200
170 INPUT "# OF MONTHS = ";N
180 IF X=1 THEN INPUT "PAYMENT= ";
PMT ELSE 220
190 PV=PMT*(1-(1+I)^(-N))/I
200 PRINT USING "PV= #####.##";
PV:PAUSE
210 GOTO 100
220 INPUT "PV= ";PV
230 PMT=PV*I/(1-(1+I)^(-N))
240 PRINT USING "PMT= #####.##";
PMT:PAUSE
250 GOTO 100
260 PRINT "BYE":PAUSE 2:
END

```

I designed the program so that data must be entered as follows:

- Enter the interest rate as stated. For example, if the newspaper states APR 2.9%, enter 2.9, not .029.
- Enter the total number of months in your contract. If financing your purchase over a term of 4 years, enter 48 as the number of months. (For you technical types, this pro-

Cartridges and Accessories

The current library of application cartridges for programmable calculators includes:

TI-74

Mathematics
Statistics
Learn Pascal
Chemical Engineering
Finance

TI-95

Chemical Engineering
Mathematics
Statistics

Accessories for the TI-74 and TI-95

8K Constant Memory Cartridge. An internal battery maintains contents when removed from the calculator.

CI-7 Cassette Interface. Allows users to store and retrieve information from any standard cassette recorder.

PC-324 Dot Matrix Thermal Printer. Can print 24 characters per line at 48 lines per minute. An adapter (AC9201) is available to provide power for both the printer and calculator.

gram assumes a standard annuity, fully amortized over the life of the term with twelve periods per annum.)

Of course, the program can also help with purchase decisions for houses, boats, trailers, or anything that requires periodic payments over a given amount of time. ■

Moving?

Please send your previous and current address to:
Programmable Calculator News
P.O. Box 53
Lubbock, TX 79408



**TEXAS
INSTRUMENTS**

P.O. Box 53
Lubbock, Texas 79408

Bulk Rate
U.S. Postage
PAID
Dallas, Texas
Permit No. 2758