C. Shaw
3/1/79

CALCULATOR CARTRIDGE SPECIFICATION (preliminary)

PURPOSE To provide the user with functions found on advanced scientific and financial calculators in a format that can be easily understood and used, with no programming experience required. Unlike calculators, which have a confusing array of keys, some performing 2 or 3 different functions, this program will require the user to type in the name of the function, e.g. "SIN". Users will be able to display a menu listing all of the functions. The more complex functions will prompt the user as to what input is required. All of the functions currently exist on - pocket calculators, but no calculator combines all of these functions and no calculator has the memory, flexibility of a computer.

DISPLAY FORMAT The display format will be similar to that found on printing calculators. The current result will be displayed at the bottom of the screen. As functions are performed, the display will roll up so that 24 lines of previous functions and results will be displayed. The option will exist of printing all or part of the calculations on the printer as well. The possibility exists of being able to plot various functions using the graphics mode, if space allows.

CARTRIDGE LOCATION This will be an 8K cartridge. There are certain advantages to making it a "B" slot cartridge. It would be able to use the sin, cos, arctan, and square root routines contained in the Basic cartridge in the "A" slot, saving about 650 bytes within the cartridge. Also, this would make the functions available to the users to call as subroutines in their Basic programs. This would considerably increase the power of Basic and enable the users to alternate between Basic and Calculator mode with ease.

On the other hand, Candy will take only "A" slot cartridges, so Basic cannot be run with another cartridge in that system. One possibility would be to produce two somewhat different cartridges. The one for Candy would have the sin, cos, arctan and square root routines built in, so it would have fewer additional functions. The one for Colleen would run with the Basic cartridge, so it could have additional power and flexibility.

Another idea is to have a separate financial cartridge containing all of the functions found on advanced financial calculators, including compound interest, annuities, linear regression, and depreciation. Most of the scientific functions would be omitted, including the trig functions.

The cartridge(s) will not require any controllers or peripherals, since all input is from the keyboard and all output is to the screen. A printer is optional.

ACCURACY Another question is that of accuracy. Major calculator manufacturers have devoted a great deal of time to ensuring that their calculators are accurate to 8 or 10 significant figures. It appears unlikely that we can maintain that kind of accuracy in complicated calculations. The cartridge will use the floating point routines provided with the Basic. These have already been shown to have two major bugs and may possibly have more. In test cases the functions were accurate to 8 or 9 significant figures, except when applying trig functions to very large angles. However, when the functions were combined as in $y^x = \exp(x * \log(y))$, accuracy dropped to 6 significant digits. Further accuracy studies are called for.

REGISTERS and STACKS

As in pocket calculators, the numbers entered will be stored on a stack until needed for a calculation. Both Reverse Polish and Algebraic notation will be allow. Switching from one notation to the other will clear all pending operations, leaving only the x register (the top of the stack) and the memory registers. A large stack depth will be implemented so that the user doesn't have to worry about stack overflow or too many open parenthesis. The y register is the location below the top of stack. One-variable functions are applied to x with the result put in x. Two-variable functions are applied to x and y; x and y are popped off the stack and f(x, y) is pushed on the stack to form the new x.

POSSIBLE FUNCTIONS AND KEYWORDS

Modes

RPN Use Reverse Polish Notation
* ALG Use Algebraic notation with operator precedence

DEG Trig. functions in Degrees
* RAD " " " Radians
GRAD " " " Grads

* DEC Results displayed in Decimal notation
HEX " " " Hexadecimal "
OCT " " " Octal "

SCI Results displayed in Scientific notation
ENG " " " Engineering " (exponent is multiple of 3)
* NOEXP " " with no exponent (if possible)

FIX x Display x digits to right of decimal point (with rounding)
* NOFIX Display full precision

MENU List all commands in alphabetical order or in groups (as in this document).

* PROMPT Provide prompting for complicated operations.
NOPROMPT Don't " " " " "
SAME In prompting mode, use same value for parameter.

* = default

Printer Commands

PRINT
- ON — Print all results and operations as in printing calculator
- OFF — Don't print.
- REGS — Print register names and contents
- ADV — Advance printer (blank line)
- X — Print current result (x-register) only
- LIST — List program (if programmable capability)

Numeric Keys

- 0-7 — all modes (OCT, DEC, HEX)
- 8-9 — DEC and HEX only
- A-F — HEX only

Hex numbers must be delimited by '#' if they contain A-F.

Numeric Format is same as Basic e.g. 1.5E-2 = .015
HEX and OCT will not allow an exponent, so use of 'E' for 14_{16} and exponent will not cause problems.

PI π 3.141592654... (as accurate as possible)

Register Control

<shft><del> Clear Entry (clears line)
- CLR — Clear entry, stack, but not mem
- CLR MEM — Clear everything
- XCHGY — x ⇄ y exchange
- <CARRIAGE RETURN> — Performs ENTER↑ function in RPN (push)
- = , (,) — Algebraic only (= closes all parens.)

NOTE: a large stack will be used to allow many nested parenthesis or many stacked numbers in RPN.

POP — Pop top of stack in RPN

Functions provided by Shapardson Basic

	+	add	$X = X + Y$
	−	subtract	$X = X - Y$
	*	multiply	$X = X * Y$
	/	divide	$X = X / Y$
*	SQRT	$\sqrt{X}$	
	LN	$\log_e X$	
	LOG10	$\log_{10} X$	
	EXP	e^X	
	EXP10	10^X	
*	SIN	} Affected by DEG, RAD, GRAD	
*	COS		
*	ARC	-(only arctan provided)	

* = in Basic cartridge, not built-in.

Additional One and Two Variable Functions

	SQUARE	$X^2 = X * X$
	POWER	Y^X = EXP10 (X * LOG10 (Y))
	ROOT	$\sqrt[X]{Y}$ = EXP10 ((1/X) * LOG10 (Y))
	INTEGER	truncate (integer part only)
	FRACTION	= X − INTEGER(X)
††	ABSVAL	$\lvert X \rvert$
	RECIP	$1/X$ (reciprocal)
	FACTORIAL	$X!$
†	PERMUTATION	$P\binom{X}{Y} = \frac{X!}{(X-Y)!}$
†	COMBINATION	$C\binom{X}{Y} = \frac{X!}{Y!(X-Y)!} = \frac{P\binom{X}{Y}}{Y!}$
†	PERCENT	%
	TAN	= SIN / COS
	ARC	e.g. ARC HYP SIN (X) = $\sinh^{-1}(X)$ = LN (X + SQRT (X * X + 1))
†	HYP	e.g. HYP SIN (X) = $\sinh(X)$ = .5 * (EXP (X) − EXP (−X))

† = prompting messages available

†† may be left out − not commonly provided on calculators

Bit manipulation (HEX and OCT only)

LSHF	shift y left x bits
RSHF	shift y right x bits
COMP	1's complement x
OR	X = x OR y
AND	X = x AND y
XOR	X = X exclusive or y

Memories

There will be 100 memory registers, numbered 00-99. Any 1-variable function may be performed on a memory register, e.g. MEM 00 SIN replaces the contents of location 00 with the sin of the contents.

STO xx	xx ← x	(xx = 00 - 99)
RCL xx	x ← xx	
SUM xx	xx = xx + x	
SUB xx	xx = xx - x	
PRD xx	xx = xx * x	
DIV xx	xx = xx / x	
MEM xx	next (1-variable) operation will apply to xx only.	
XCH xx	Exchange x and xx	x ⇄ xx

Programmability

Extensive user programmability is judged to be unnecessary because BASIC provides that capability. However, it may be desirable to have a program memory to allow simple repetitive calculations. No branch, test, or editing operations are planned.

PROGRAM	Store following steps in program memory
END	Stop storing in program mem.
RESET	Go to beginning of user program.
RUN	Execute user program
STOP	In program, stop until RUN entered by user. From keyboard, halt program.
PAUSE	Brief pause during program execution.

Conversions – 1-variable

† CONV <old units> < new units>

DMS	Degrees Minutes Seconds	DD.MMSSSS (fractional seconds) can be used for hr-min-sec
DECDEG	Decimal Degrees	DD.DDDD
DATE	MMDD.YYYY Format	
DECDATE	No. of day since start of Gregorian Calendar. User to compute days between dates.	

Length

INCHES
FEET
YARDS
MILES
CM
METERS
KM
NAUT Nautical Miles

Temperature

FAHR °F
CENT °C

Mass

OZ Ounces
POUNDS
GRAMS
KG

Volume

TSP Teaspoons
TBSP Tablespoons
CUPS
QUARTS
GAL Gallons
FLOZ Fluid Ounces
LITERS
CC Cubic Centimeters

Conversions can be made from any member of a group to any other units contained in the same group

e.g. X reg = 2.54 CONV CM INCHES
result: X reg = 1

† = prompting provided

C. Shaw
p. 11
3/1/79

Conversions: 2-variable

† CONV

POLAR	$x = R,\ y = \Theta$
RECT	$x,\ y$

† = prompting available

Multivariable Functions and Other Functions

These functions usually require that more than two number be entered by the user. All provide prompting. Menu select feature: User can display menu of multivariable functions and select one by number.

Statistics

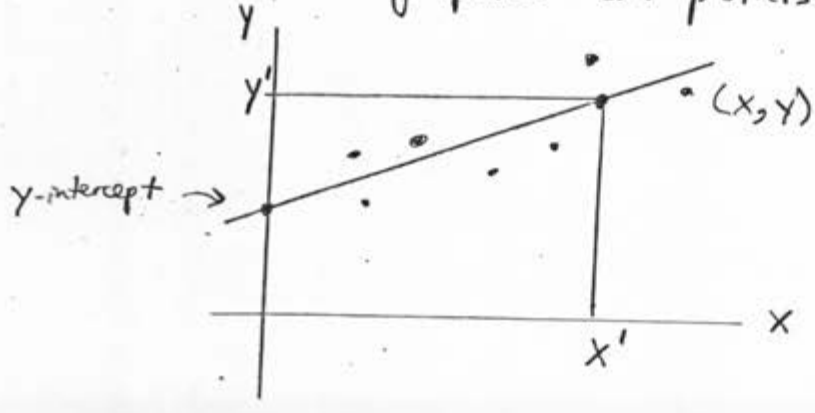
Linear Regression

User types LINEAR REGRESSION or uses menu select. Then user enters pairs of points (x, y). Following functions may be computed:

MEAN	of x- and y- arrays
STDDEV	Standard deviation of x- and y- arrays N and N-1 weighting are possible.
VARIANCE	of x- and y- arrays
YINT	Y- Intercept } line passing thru points (best fit)
SLOPE	
CORRELATION	Correlation Coefficient
ALL	Compute and display all of the above
YPRIME	compute Y' for new x
XPRIME	compute x' for new y

User may enter x's only, with no y's, to compute STDDEV and VARIANCE.

PLOT Draw graph of all points, best fit line.



Polynomial Evaluation

This function is defined and used by BASIC.

Compute $P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$

Plot $P(x)$ for specified range of x.

Day of the Week, Days Between Dates

DAY Compute day of the week for x = MM DD. YYYY

Also provides prompting for computing Days between Dates and date x days before or after date y.

Takes into account number of days in month and leap year.

Functions used:

DECDATE (MMDD. YYYY) =

Jan. and Feb:

365 (YYYY) + DD + 31 (MM-1) + INT[(YYYY-1)/4]
- INT (3/4 [INT (YYYY-1) /100+1])

Mar - Dec:

365 (YYYY) + DD + 31 (MM-1) - INT (.4 MM+2.3)
+ INT (YYYY/4) - INT (3/4 [INT (YYYY/100)+1])

Day of week = DECDATE + INT (-DECDATE / 7 x 7]

Returns number from 0-6 which is converted to word as follows:
0 = SAT 1 = SUN 2 = MON 3 = TUES 4 = WED 5 = THURS
6 = FRI

(From TI 58/59 Master Library Manual, p. 76)

Random Numbers

UNIF Uniform Distribution
LL enter lower limit
UL enter upper limit
RND generate random number, put in x reg.

NORM Normal Distribution
MEAN enter desired mean $\bar{x}$ (from x reg)
STDDEV enter desired standard deviation σ (from x reg)
RND generate random number, put in x reg.

Default is same as for BASIC.

Compound Interest

FV	Future Value
PV	Present Value
I	Interest rate per period (in %)
i	I / 100 = Interest rate per period $0 \le i \le 1$
N	number of periods
R	annual interest rate (in %)
r	R / 100 = annual interest rate $0 \le r \le 1$
q	periods per year
n	number of years

$$FV = PV(1+i)^n = PV\left(1+\frac{r}{q}\right)^{nq}$$

$$i = r/q \qquad I = R/q$$
$$N = nq$$

User enters n known variables, computer returns with single unknown variable.

Newton-Raphson iteration used to compute i.

Annuities

Same variables as Compound Interest.

In addition:

PMT — fixed payment made at either beginning or end of each period.

BAL — Balloon payment: lump sum paid at end of term. (Optional)

* $FV = PMT \times \frac{(1+i)^N - 1}{i} + BAL$ (Sinking Fund)

† $FV = PMT \times (1+i) \times \frac{(1+i)^N - 1}{i} + BAL$

* $PV = PMT \times \left[\frac{1-(1+i)^{-N}}{i} \right] + \left[BAL \times (1+i)^{-N} \right]$

† $PV = PMT \times (1+i) \times \left[\frac{1-(1+i)^{-N}}{i} \right] + \left[BAL \times (1+i)^{-N} \right]$

* = pay at end of payment period (ordinary annuity)
† = " " beginning " " " (annuity due)

(From TI 58/59 Master Library Manual, p. 67).

The instruction book for the cartridge will explain the meanings and uses of these formulas.

Other Possible Functions

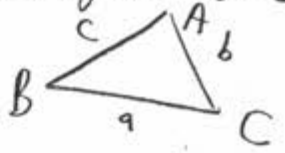
If space allows, which seems unlikely, other functions may be added.

Complex Arithmetic: $x = a + b\,i$, $+, \div, \times, \div, y^x$, etc.

Matrix Arithmetic; Determinants, Simultaneous Equations

Find Zeros of Function

Triangle Solution: Given ASA, SSS, etc. compute other parts of triangle.



Curve Solution:

Given (θ, r), (θ, s), (θ, c), (r, s), or (r, c) compute other parameters.

Etc., etc.

Financial Functions

DDB	Double Declining Balance	} Depreciation
SOYD	Sum of Years Digits	
SL	Straight Line	

Compute purchase price, Interest rates on bonds and notes.

Evaluate company financial statement.

Etc., etc.

C. Shaw
P. 18
3/1/79

Display Example

Display		Printer	
	DEG		
5	5		
3	3		
2	2		
	*** SIN		
.0348994	.0348994		
	*** ARC SIN		
2	2		
	***		
	PRINT X	2	PRINT X
	PRINT ON		
	POWER		POWER
3	3	3	3
8	***	8	***
5	5	5	5
40	***	40	***
	PRINT OFF		
28	HEX		
50	OCT.		

*** indicates result displayed by computer.

The display scrolls upward, so the most recent 24 lines are displayed, including the current entry line at the bottom.

Example of Prompting:

		PROMPT
		CONV
		POLAR } follow each entry by a carriage return
		RECT
		DEG
enter current units		
enter new units		
enter angle mode		
enter angle theta	45	
enter radius R	5	
y =	3.5355339	*Y*
x =	3.5355339	*X*
enter current units		
		NOPROMPT
		DEG
		RPN
	3.5355339	ENTER
	3.5355339	
		CONV RECT POLAR
	45	*Y*
	5	*X*

A series of computations may be performed for each entry. A carriage return enters whatever is in the x register. To force carriage return to have the ENTER↑ rather than parameter entry function when in RPN and PROMPT modes, type <esc> <carriage return>. The program will save previously entered parameters so that the user doesn't have to worry about whether the stack has been altered or not when in PROMPT mode. To use the same value over again for a certain parameter, type "SAME" and the previous value will be displayed.

Memory Requirements

The ROM required by each function depends on the algorithm used and the amount of prompting provided. ASCII characters take up a lot of memory, so an efficient means of storing all of the messages must be devised, possible using some sort of encoding.

Calculator Comparisons

I plan to start with a four-function calculator program and add the most desirable functions until I run out of room. Probably the compound interest and annuity formulas will be included, with other financial functions to be included in a separate financial cartridge. The first cartridge will contain roughly the same functions as the TI-55 advanced scientific calculator with statistical functions and simple programmability (Best's $34.93) plus some of the functions found in the Master Library Module ($35) of the TI-58 ($88.96 at Best's) plus all of the hexadecimal and octal capability of the TI Programmer (≈ $60). The accuracy of the cartridge will sometimes be less than that of the calculators but should be adequate for most applications. The cartridge should be easier to use than the pocket calculators. It has the advantage of being able to display many characters at once on the 40 column screen and being able to print them on the 40 column printer. The lack of a 10-key keyboard with digits 0-9 may be seen as a disadvantage.* In financial calculators, we will be competing with the TI MBA, HP 37E preprogramed financial ($64.96), the HP-92 desktop printer equivalent of the HP 37E (several 100 dollars), the HP-38E keystroke programmable financial ($104.97) and a host of others.

* But not to those who can touch type on a typewriter, not a 10-key.

Manual This cartridge will require more in the way of instructions than previous cartridges. Users would undoubtedly like to have examples and explanations of the derivations and applications of each function in addition to a summary of the commands for reference purposes. For example, they will want to know how they can compare loans by using the compound interest and annuity formulas.

Suggestions

Any suggestions for functions to be added or omitted, or for other improvements, will be appreciated. This document is preliminary, so changes will be made and further details will be worked out.