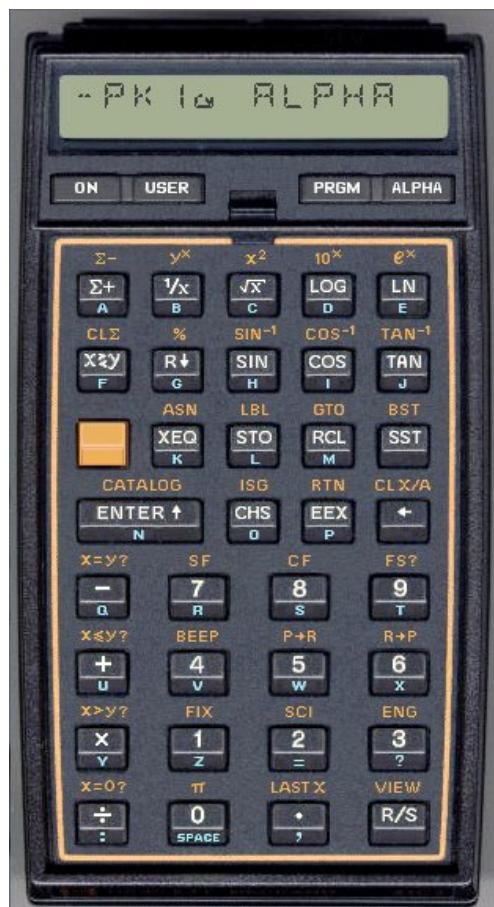


PK Collection Manual

February/March 2015



Modules for the HP41



Preface

The Story of the PK Collection is this:

I bought my HEPAX in 1988 and also got Ken Emery's book **HP-41 MCODE FOR BEGINNERS**. I soon got into mcode programming and Emery's book was a tremendous help. Many of my programs came from ideas I got from this book and also from the **VASM** listings. Many programs are there just because I wanted to test the techniques and not so much because they had a purpose (like it's not very often you would need a tripple test $X=YZT$? But it was fun programming it. The programs **JUMP1** and **JUMP3** are in the HEPAX manual, but as FOCAL. I used the *partial key sequence* to make an mcode version that has a fancier input, but basically does the same thing).

In 1990 I bought some **ZEPROMs** and put my programs into actual Modules.

They were never meant to be published.

Around 2000 I wanted to optimize the contents of my modules, but this turned out to be a bad idea. Due to lack of time and energy, I never got the project finished, and I managed to get the internal code mixed up, because I wanted to change the XROM ID's and move functions to other modules. The modules were then put back in the drawer.

In summer of 2014 Ángel Martin asked the HP Community to contribute all the ROMs existing to be used for the CL project. I sent him the FORECASTER ROMs and some others that I thought he might not have. And I also sent him some of my own ROMs, not to be part of the CL, but because I thought he might appreciate some of the techniques because he is the mcode master of the 21st Century.

Angél thought that the ROMs were so good that they should be published. I said OK, but it turned out, that the XROM ID mix-up problems were rather massive, and as I at that time was trying to recover the history of my family, I put it in the back of my head.

In february 2015 Ángel made a renewed request, and this time I took the bait.

As Ángel pointed out, this is pretty much like Geneology: noone remembers anything and the documents are nonexistent or very hard to read and understand ☺.

We have now fixed the ID thing and the modules are OK (we think), and I have done this manual as well. If anybody can use it, that's fine!

You will probably find, that there are still bugs in the programs and flaws in the manual - it's been quite a task to memorize 25 years back what to input and what output to expect.

A tremendous **thanks** goes to Ángel who has put a lot of work into this (but then, it was his idea ☺)

A number of the programs are not my own. I have tried to credit the original writers, and if I missed someone it is not intentional. Apologies for the inconvenience of some programs written in Danish ☺. I have tried to compensate by writing this manual in english...(more or less)

Fredericia, Denmark, March 2015

Poul Kaarup

The Modules

PK1 8K	-PK1a ALPHA -PK1a POINT -PK1a REG'S	PK2 8K	-PK2c MATH	PK3 8K	-PK3e TIME -PK3e CONV -PK3e REG -PK3e UTIL
	-PK1b FLAGS -PK1b STK -PK1b TONE		-PK2d PHYS		-PK3f PRGM

Contents

- Page 4 Alpha functions
- Page 9 Pointer functions
- Page 10 Register functions
- Page 11 Flag functions
- Page 14 Stack functions
- Page 16 Tone functions
- Page 17 Table functions
- Page 20 Math functions
- Page 39 Physics functions
- Page 44 Time functions
- Page 46 Conversion functions
- Page 47 Utility functions
- Page 48 Programming functions
- Page 56 Formulas used in the programs
- Page 60 Technical stuff
- Page 61 XROM numbers

Credits

Ken Emery	DISPLAY, LOOP, 10-BASE, RSLCT, Y<>Z, QR, X=YZ?, A>REG, REG>A
Clifford Stern	DEBUG, HEXENTRY
ASSEMBLER3	APPFN, CSST, UPFAT, APPLBL, GE, GORAM, XQ>XR
CCD/W&W	ABSP, ARCLI, CLA-, KEY?, PMTK, PMTA, UPDFAT
McCornack/Jarrett	PPACK, XTO3N, 3NTOX, XTON, NTOX, INCRM, DECRM
Claus Primdahl Nielsen	FLAGMAP
Kim Holm	CP2, CPF
Ángel Martin	MLOAD, ADOW, DDATE
Hewlett Packard	P<>S, DSE, ISZ

PK1a

-PK1a ALPHA

Header ♦ Alpha functions

ADL

Alpha Delete Left

Deletes the leftmost character in ALPHA

	X	Y	Z	T	L	ALPHA
Before						ALPHA
After						LPHA

ADR

Alpha Delete Right

Deletes the rightmost character in ALPHA

	X	Y	Z	T	L	ALPHA
Before						ALPHA
After						ALPH

ADX

Alpha Delete Xth

Deletes the Xth character in ALPHA

	X	Y	Z	T	L	ALPHA
Before	2					ALPHA
After	2					APHA

AXL

Alpha to X Left

Puts the leftmost character in ALPHA to X

	X	Y	Z	T	L	ALPHA
Before						ALPHA
After	65	X	Y	Z		LPHA

AXR

Alpha to X Right

Puts the rightmost character in ALPHA to X

	X	Y	Z	T	L	ALPHA
Before						ALPHA
After	65	X	Y	Z		ALPH

AXX

Alpha to X Xth

Puts the Xth character in ALPHA to X * NEW X and not the OLD saved in LASTX

	X	Y	Z	T	L	ALPHA
Before	2					ALPHA
After	76	X	Y	Z	76*	APHA

XAL

X to Alpha Left

Appends the character in X to ALPHA on the left

	X	Y	Z	T	L	ALPHA
Before	88					ALPHA
After	Y	Z	T	0		XALPHA

XAR

X to Alpha Right

Appends the character in X to ALPHA on the right

	X	Y	Z	T	L	ALPHA
Before	88					ALPHA
After	Y	Z	T	0		ALPHAX

XAY

X to Alpha Yth

Appends the character in X to ALPHA on the Yth place

	X	Y	Z	T	L	ALPHA
Before	88	2				ALPHA
After		Z	T	0		A X ALPHA

XSY	X Substitute Yth	Substitutes the character in X with the Yth character in ALPHA									
	X	Y	Z	T	L	ALPHA					
Before	88	2				ALPHA					
After	76	Z	T	0		A X PHA					
XAI	X TO ALPHA INTEGER	Appends integer part of X to ALPHA rightmost									
	X	Y	Z	T	L	ALPHA					
Before	9					ALPHA					
After	9					ALPHA9					
YAI	Y TO ALPHA INTEGER	Appends integer part of Y to ALPHA rightmost									
	X	Y	Z	T	L	ALPHA					
Before		9				ALPHA					
After		9				ALPHA9					
A=X	ALPHA = X	Appends = and X to ALPHA and displays it (uses the current digit format)									
	X	Y	Z	T	L	ALPHA					
Before	9					ALPHA					
After	9					ALPHA= 9.0000					
A=Y	ALPHA = Y	Appends = and Y to ALPHA and displays it (uses the current digit format)									
	X	Y	Z	T	L	ALPHA					
Before		9				ALPHA					
After		9				ALPHA= 9.0000					
VWA	VIEW ALPHA	Displays/prints ALPHA									
	<i>* Used by a number of programs in this collection. If you use V41 or another emulator you may experience that VWA doesn't halt the program as expected.</i>										
	<i>Solution: turn calculator OFF and then ON to clear flag 55</i>										
VWAP	VIEW ALPHA PAUSE	Displays ALPHA and pauses the program									
VWT	VIEW AS TIME	Appends X to ALPHA as time and displays it (uses the current digit format)									
	X	Y	Z	T	L	ALPHA					
Before	4,5678					ALPHA					
After	4,5678					ALPHA 4:56:78					
VW%	VIEW AS %	Appends X·100 and % to ALPHA (uses the current digit format)									
	X	Y	Z	T	L	ALPHA					
Before	0.88					ALPHA					
After	0.88					ALPHA88.00 %					
A(X)=Y	ALPHA (X) = Y	Displays ALPHA (X) = Y (appends X as integer and uses current digit format for Y)									
	X	Y	Z	T	L	ALPHA					
Before	4.00	3.00				ALPHA					
After	4.00	3.00				ALPHA(4) = 3.00					

A(XY)=Z	ALPHA (X,Y) = Z	Displays ALPHA (X,Y) = Z (appends X and Y as integers and uses current digit format for Z)																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>4.00</td><td>3.00</td><td>2.00</td><td></td><td></td><td>ALPHA</td></tr> <tr> <td>After</td><td>4.00</td><td>3.00</td><td>2.00</td><td></td><td></td><td>ALPHA(4,3) = 2.00</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	4.00	3.00	2.00			ALPHA	After	4.00	3.00	2.00			ALPHA(4,3) = 2.00
	X	Y	Z	T	L	ALPHA																	
Before	4.00	3.00	2.00			ALPHA																	
After	4.00	3.00	2.00			ALPHA(4,3) = 2.00																	
A(IX)=Y	ALPHA INDIRECT X = Y	Same as A(X)=Y but uses X as a pointer (uses the current digit format for Y)																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>R05</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>5.00</td><td>3.00</td><td></td><td></td><td>23.00</td><td>ALPHA</td></tr> <tr> <td>After</td><td>5.00</td><td>3.00</td><td></td><td></td><td></td><td>ALPHA(23)=3.00</td></tr> </tbody> </table>		X	Y	Z	T	R05	ALPHA	Before	5.00	3.00			23.00	ALPHA	After	5.00	3.00				ALPHA(23)=3.00
	X	Y	Z	T	R05	ALPHA																	
Before	5.00	3.00			23.00	ALPHA																	
After	5.00	3.00				ALPHA(23)=3.00																	
A=(XY)	ALPHA=(XxY)	Displays X and Y as a pair of numbers appended to ALPHA																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>1.00</td><td>2.00</td><td></td><td></td><td></td><td>ALPHA</td></tr> <tr> <td>After</td><td>1.00</td><td>2.00</td><td></td><td></td><td></td><td>ALPHA=(1.00x2.00)</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	1.00	2.00				ALPHA	After	1.00	2.00				ALPHA=(1.00x2.00)
	X	Y	Z	T	L	ALPHA																	
Before	1.00	2.00				ALPHA																	
After	1.00	2.00				ALPHA=(1.00x2.00)																	
X Y	X Exponent Y	Displays X and Y as a number and its exponent (uses the current digit format for X)																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>2.00</td><td>3.00</td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>2.00</td><td>3.00</td><td></td><td></td><td></td><td>2.00 E 3</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	2.00	3.00					After	2.00	3.00				2.00 E 3
	X	Y	Z	T	L	ALPHA																	
Before	2.00	3.00																					
After	2.00	3.00				2.00 E 3																	
STAK	STAK NAMES	Writes the names of the registers to the stak and ALPHA																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>X</td><td>Y</td><td>Z</td><td>T</td><td>L</td><td>ALPHA</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before							After	X	Y	Z	T	L	ALPHA
	X	Y	Z	T	L	ALPHA																	
Before																							
After	X	Y	Z	T	L	ALPHA																	
TLC	TOGGLE LOWER CASE	Toggles whether ALPHA writes upper or lower case (HP41CX only)																					
A>REG	ALPHA TO REGISTERS	Saves the contents of ALPHA into registers 00-03 as NonNormalisedNumber *																					
		<table border="1"> <thead> <tr> <th></th><th>R00</th><th>R01</th><th>R02</th><th>R03</th><th></th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td>ABCDEFGHIJKLMNPQRSTUVWXYZ</td></tr> <tr> <td>After</td><td>NNN</td><td>NNN</td><td>NNN</td><td>NNN</td><td></td><td>ABCDEFGHIJKLMNPQRSTUVWXYZ</td></tr> </tbody> </table>		R00	R01	R02	R03		ALPHA	Before						ABCDEFGHIJKLMNPQRSTUVWXYZ	After	NNN	NNN	NNN	NNN		ABCDEFGHIJKLMNPQRSTUVWXYZ
	R00	R01	R02	R03		ALPHA																	
Before						ABCDEFGHIJKLMNPQRSTUVWXYZ																	
After	NNN	NNN	NNN	NNN		ABCDEFGHIJKLMNPQRSTUVWXYZ																	
		* If you try to RCL or VIEW the registers 00-03 they will be normalised and REG>A will not work properly																					
REG>A	REGISTERS TO ALPHA	Rebuilds ALPHA from the contents in registers 00-03 *																					
		<table border="1"> <thead> <tr> <th></th><th>R00</th><th>R01</th><th>R02</th><th>R03</th><th></th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>NNN</td><td>NNN</td><td>NNN</td><td>NNN</td><td></td><td></td></tr> <tr> <td>After</td><td>NNN</td><td>NNN</td><td>NNN</td><td>NNN</td><td></td><td>ABCDEFGHIJKLMNPQRSTUVWXYZ</td></tr> </tbody> </table>		R00	R01	R02	R03		ALPHA	Before	NNN	NNN	NNN	NNN			After	NNN	NNN	NNN	NNN		ABCDEFGHIJKLMNPQRSTUVWXYZ
	R00	R01	R02	R03		ALPHA																	
Before	NNN	NNN	NNN	NNN																			
After	NNN	NNN	NNN	NNN		ABCDEFGHIJKLMNPQRSTUVWXYZ																	
A A	ALPHA <> ALPHA	Swaps two words in ALPHA separated by a space																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td>ALPHA BETA</td></tr> <tr> <td>After</td><td></td><td></td><td></td><td></td><td></td><td>BETA ALPHA</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before						ALPHA BETA	After						BETA ALPHA
	X	Y	Z	T	L	ALPHA																	
Before						ALPHA BETA																	
After						BETA ALPHA																	
ANL	ALPHA NUMBER LEFT	Puts a number from the left of ALPHA into X (uses current digit format)																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td>3.00 ALPHA</td></tr> <tr> <td>After</td><td>3.00</td><td>X</td><td>Y</td><td>Z</td><td>L</td><td>ALPHA</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before						3.00 ALPHA	After	3.00	X	Y	Z	L	ALPHA
	X	Y	Z	T	L	ALPHA																	
Before						3.00 ALPHA																	
After	3.00	X	Y	Z	L	ALPHA																	

ANR	ALPHA NUMBER	Puts a number from the right of ALPHA into X (uses current digit format)																					
	RIGHT																						
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td>ALPHA 3.00</td></tr> <tr> <td>After</td><td>3.00</td><td>X</td><td>Y</td><td>Z</td><td>L</td><td>ALPHA</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before						ALPHA 3.00	After	3.00	X	Y	Z	L	ALPHA
	X	Y	Z	T	L	ALPHA																	
Before						ALPHA 3.00																	
After	3.00	X	Y	Z	L	ALPHA																	
ALF	ALPHABETIZED	Puts the characters in ALPHA in alphabetic order																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td>ALPHA</td></tr> <tr> <td>After</td><td></td><td></td><td></td><td></td><td></td><td>AAHLP</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before						ALPHA	After						AAHLP
	X	Y	Z	T	L	ALPHA																	
Before						ALPHA																	
After						AAHLP																	
A<>	ALPHA REVERSED	Reverses ALPHA																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td>ALPHA</td></tr> <tr> <td>After</td><td></td><td></td><td></td><td></td><td></td><td>AHPLA</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before						ALPHA	After						AHPLA
	X	Y	Z	T	L	ALPHA																	
Before						ALPHA																	
After						AHPLA																	
CON	CONTINUE ?	Prompts CON_. Accepts only J or N. J returns 1 and N returns 2 to X																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>1 or 2</td><td>Y</td><td>Z</td><td>T</td><td></td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before							After	1 or 2	Y	Z	T		
	X	Y	Z	T	L	ALPHA																	
Before																							
After	1 or 2	Y	Z	T																			
JN	JA ELLER NEJ ?	Prompts JN_. Accepts only J or N. J returns 1 and N returns 2 to X																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>1 or 2</td><td>Y</td><td>Z</td><td>T</td><td></td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before							After	1 or 2	Y	Z	T		
	X	Y	Z	T	L	ALPHA																	
Before																							
After	1 or 2	Y	Z	T																			
GO	GAME OVER	Displays GAME OVER																					
1L	Linear Equation	Displays X and Y as a linear equation (uses the current digit format)																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>1</td><td>2</td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>1</td><td>2</td><td></td><td></td><td></td><td>Y=1.0X+2.0</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	1	2					After	1	2				Y=1.0X+2.0
	X	Y	Z	T	L	ALPHA																	
Before	1	2																					
After	1	2				Y=1.0X+2.0																	
2L	Quadratic Equation	Displays Reg's 00 to 02 as a quadratic equation (uses the current digit format)																					
		<table border="1"> <thead> <tr> <th></th><th>R00</th><th>R01</th><th>R02</th><th></th><th></th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>1</td><td>2</td><td>3</td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>1</td><td>2</td><td>3</td><td></td><td></td><td>Y=3.0X²+2.0X+1.0</td></tr> </tbody> </table>		R00	R01	R02			ALPHA	Before	1	2	3				After	1	2	3			Y=3.0X ² +2.0X+1.0
	R00	R01	R02			ALPHA																	
Before	1	2	3																				
After	1	2	3			Y=3.0X ² +2.0X+1.0																	
3L	Cubic Equation	Displays Reg's 00 to 03 as a cubic equation (uses the current digit format)																					
		<table border="1"> <thead> <tr> <th></th><th>R00</th><th>R01</th><th>R02</th><th>R03</th><th></th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>1</td><td>2</td><td>3</td><td>4</td><td></td><td></td></tr> <tr> <td>After</td><td>1</td><td>2</td><td>3</td><td>4</td><td></td><td>Y=4.0X³+3.0X²+2.0X+1.0</td></tr> </tbody> </table>		R00	R01	R02	R03		ALPHA	Before	1	2	3	4			After	1	2	3	4		Y=4.0X ³ +3.0X ² +2.0X+1.0
	R00	R01	R02	R03		ALPHA																	
Before	1	2	3	4																			
After	1	2	3	4		Y=4.0X ³ +3.0X ² +2.0X+1.0																	
DISPLAY	DISPLAY TEST	Tests the Display segments																					
		Shows 																					
		followed by 																					

VNC

VERNAM CODING

Codes a textfile in XM * uses CCD

Displays		
LBL A	Creates text-file	
	Prompts REG'S ?	Input size of text-file
	Prompts FIL_	Input filename
LBL B	Purges text-file	
	Prompts FIL_	Input filename
LBL b	Clears text-file	
	Prompts FIL_	Input filename
LBL C	Codes text-file	
	Prompts RNDM?	Input (& remember!) random number *
	Prompts FIL_	Input filename
LBL c	Codes ALPHA	
	Prompts TEXT: _	Input (& remember!) random number *
	Displays 	Input text to be coded or similar stores the text in R ₀₁ -R ₀₄
LBL D	Decodes text-file	
	Prompts RNDM?	Input random number (must be same as LBL C!)
	Prompts FIL_	Input filename
LBL d	Decodes ALPHA	
	Prompts RNDM?	Input random number (must be same as LBL c!)
	Displays TEXT	or similar The coded text is still in R ₀₁ -R ₀₄
LBL E	Edits text-file	
	Prompts FIL_	Input filename
	Input text to be coded or view coded/decoded text	
LBL J	Displays menu	

* Random numbers can be different for different text-files, but must be the same for coding/decoding

-PK1a POINT

Header ♦ Pointer functions

BPT

Build Pointer

Puts together a pointer used in other functions (format yyy,xxx)

If flag 00 clear:

	X	Y	Z	T	L	ALPHA
Before	3	2				
After	2,003	2			3	

If flag 00 set (flag 00 is cleared):

	X	Y	Z	T	L	ALPHA
Before	2					
After	1,002				2	

BPTI

Build Pointer with Increment

Puts together a pointer used in other functions (format zzz,yyyxx)

	X	Y	Z	T	L	ALPHA
Before	4	3	2			
After	2,00304	3	2		4	

BPTR

Build Pointer Relative

Puts together a pointer used in other functions (format xxx+1,yyy+xxx)

If flag 00 clear:

	X	Y	Z	T	L	ALPHA
Before	3	2				
After	4,005	2			3	

If flag 00 set (flag 00 is cleared): Pointer is also stored in IND X.

CLI

Clear Increment

Clears the increment part of a pointer

	X	Y	Z	T	L	ALPHA
Before	2,00304					
After	2,003					

PTR

Pointer Reverse

Reverses pointer

	X	Y	Z	T	L	ALPHA
Before	2,003					
After	3,002					

-PK1a REG'S		Header ♦ Register functions				
STO	STORE	Extended STO	Prompts STO ____	*		
ST+	STORE +	Extended ST+	Prompts ST+ ____	*		
ST-	STORE -	Extended ST-	Prompts ST- ____	*		
ST/	STORE /	Extended ST/	Prompts ST/ ____	*		
ST*	STORE *	Extended ST*	Prompts ST* ____	*		
RCL	RECALL	Extended RCL	Prompts RCL ____	*		
RC+	RECALL +	Recalls and adds to X the value in ____		Prompts RC+ ____	*	
RC-	RECALL -	Recalls and subtracts from X the value in ____		Prompts RC- ____	*	
RC/	RECALL /	Recalls and divides X with the value in ____		Prompts RC/ ____	*	
RC*	RECALL *	Recalls and multiplies X with the value in ____		Prompts RC* ____	*	
X<>	X SWAP	Extended X<>	Prompts X<> ____	*		
VIEW	VIEW	Extended VIEW	Prompts VIEW ____	*		
<i>* works for all registers, but not the stack registers. Not Programable!</i>						
ST>L	STORE INDIRECT L	Stores the value in X into the register pointed by LASTX. Increments LASTX				
			X	R25	L	ALPHA
		Before	3,14		25	
		After	3,14	3,14	26	
ST>0	STORE INDIRECT 0	Stores the value in X into the register pointed by R00. Increments R00				
			X	R00	R25	ALPHA
		Before	3,14	25		
		After	3,14	26	3,14	
RC>L	RECALL INDIRECT L	Recalls the value in the register pointed by LASTX. Increments LASTX				
			X	Y	Z	R25
		Before	X	Y	Z	3,14
		After	3,14	X	Y	3,14
RC>0	RECALL INDIRECT 0	Recalls the value in the register pointed by R00. Increments R00				
			X	Y	Z	R00
		Before	X	Y	Z	25
		After	3,14	X	Y	26
P<>S	PRIMARY SWAP SECONDARY	Swaps the contents of R00-R09 with that of R10-R19				
DSZ	DECREMENT AND SKIP ON ZERO	Decrements the value in R00, and skips the next program line if the integer part is zero. <i>A positive start-value in R00 is expected</i>				
ISZ	INCREMENT AND SKIP ON ZERO	Increments the value in R00, and skips the next program line if the integer part is zero. <i>A negative start-value in R00 is expected</i>				
PAPER	ROLL THE PAPER OUT	Equal to 4 pushes of the Paper Out button on the Printer				
?	ADD ? AND PROMPT	Adds ? to ALPHA and prompts for input				

FLAGMAP

FLAGMAP

Prints or displays which system flags are set

If Printer is present and ON it prints a matrix of the 56 system flags and marks the flags set:

```
'' FLAGMAP ''
  0 1 2 3 4 5 6 7 8 9
0 | | | | | | | | | |
1 | | | | | | | | | |
2 | | * | | | | | * | * | *
3 | | | | | | | | * | |
4 | * | | | | | | | | |
5 | | | | | | | | * |
```

If Printer is not present or not ON it displays a list of the flags set:

FLAGS SET:

22 26 28 29

37 40 50

F0 FLAGSET 0 Sets the following system flags: (SF00, USER mode, Digit grouping, DMY, FIX/ENG 4)

```
'' FLAGMAP ''
  0 1 2 3 4 5 6 7 8 9
0 | * | | | | | | | | |
1 | | | | | | | | | |
2 | | * | | | | | * | * | *
3 | | * | | | | | * | |
4 | * | * | | | | | | |
5 | | | | | | | | * |
```

F1 FLAGSET 1 Sets the following system flags: (USER mode, Digit grouping, DMY, FIX/ENG 4)

```
'' FLAGMAP ''
  0 1 2 3 4 5 6 7 8 9
0 | | | | | | | | | |
1 | | | | | | | | | |
2 | | * | | | | | * | * | *
3 | | * | | | | | * | |
4 | * | * | | | | | | |
5 | | | | | | | | * |
```

F2 FLAGSET 2 Sets the following system flags: (SF00, SF01, SF02, SF03, SF04, USER mode, Digit grouping, DMY, FIX/ENG 2)

```
'' FLAGMAP ''
  0 1 2 3 4 5 6 7 8 9
0 | * | * | * | * | | | | |
1 | | | | | | | | | |
2 | | * | | | | | * | * | *
3 | | * | | | | | * | |
4 | * | * | | | | | | |
5 | | | | | | | | * |
```

F3 FLAGSET 3 Sets the following system flags: (Range error ignore, Error ignore, Digit grouping, DMY, FIX/ENG 4)

```
'' FLAGMAP ''
  0 1 2 3 4 5 6 7 8 9
0 | | | | | | | | | |
1 | | | | | | | | | |
2 | | * | | | * | * | * | |
3 | | * | | | | | * | |
4 | * | * | | | | | | |
5 | | | | | | | | * |
```

F4	FLAGSET 4	Sets the following system flags: (Range error ignore, Error ignore, , USER mode, Digit grouping, DMY, ENG 9, Radians)																		
		<pre>'' FLAGMAP '' 0 1 2 3 4 5 6 7 8 9 0 1 2 + + + + + 3 + + + 4 + + 5 + </pre>																		
F5	FLAGSET 5	Sets the following system flags: (SF00, SF01, SF02, SF03, SF04, SF05, SF06, SF07, SF08, SF09, SF10, SF18, Range Error Ignore, Error Ignore, USER mode, Digit grouping, DMY, FIX 2)																		
		<pre>'' FLAGMAP '' 0 1 2 3 4 5 6 7 8 9 0 + + + + + + + + 1 + + 2 + + + + 3 + + 4 + 5 + </pre>																		
F6	FLAGSET 6	Sets the following system flags: (USER mode, . decimal, Digit grouping, DMY, FIX 1)																		
		<pre>'' FLAGMAP '' 0 1 2 3 4 5 6 7 8 9 0 1 2 + + + + 3 + + 4 + 5 + </pre>																		
F7	FLAGSET 7	Sets the following system flags: (SF03, USER mode, Digit grouping, DMY, FIX 2)																		
		<pre>'' FLAGMAP '' 0 1 2 3 4 5 6 7 8 9 0 + 1 2 + + + + 3 + + 4 + 5 + </pre>																		
F8	FLAGSET 8	Sets the following system flags: (SF10, SF20, Digit grouping, DMY, FIX/ENG 2)																		
		<pre>'' FLAGMAP '' 0 1 2 3 4 5 6 7 8 9 0 1 + 2 + + + + 3 + + 4 + + 5 + </pre>																		
F9	FLAGSET 9	Sets the following system flags: (SF00, FIX 0)																		
		<pre>'' FLAGMAP '' 0 1 2 3 4 5 6 7 8 9 0 + 1 2 + + 3 4 + 5 + </pre>																		
F>X	FLAGS TO X	Status of the flags 00-07 is recalled to X as a number from 0 to 255. The flags are weighted as this:																		
		<table border="1"> <thead> <tr> <th>Flag</th><th>F00</th><th>F01</th><th>F02</th><th>F03</th><th>F04</th><th>F05</th><th>F06</th><th>F07</th></tr> </thead> <tbody> <tr> <td>Value</td><td>1</td><td>2</td><td>4</td><td>8</td><td>16</td><td>32</td><td>64</td><td>128</td></tr> </tbody> </table>	Flag	F00	F01	F02	F03	F04	F05	F06	F07	Value	1	2	4	8	16	32	64	128
Flag	F00	F01	F02	F03	F04	F05	F06	F07												
Value	1	2	4	8	16	32	64	128												
X>F	X TO FLAGS	Flags 00-07 are set according to the number in X using the values above																		

FC?S	FLAG CLEAR ? AND SET	Tests if the flag in X is Clear and then sets it. Skips the next program line if test is false
FS?S	FLAG SET ? AND SET	Tests if the flag in X is Set and then sets it. Skips the next program line if test is false
CFX	CLEAR FLAG IN X	Clears the flag in X. Works with all flags
SFX	SET FLAG IN X	Sets the flag in X. Works with all flags
TFX	TOGGLE FLAG IN X	Inverts the status of the flag in X. Works with all flags
ENG	ENGINEERING MODE	Sets the Engineering mode for displaying numbers
FIX	FIXED MODE	Sets the Fixed mode for displaying numbers
SCI	SCIENTIFIC MODE	Sets the Scientific mode for displaying numbers
F/E	FIXED/ ENGINEERING MODE	Sets the Fixed/Engineering mode for displaying numbers. This means numbers larger than 999999999 will be displayed using ENG rather than SCI
FIX/ENG	FIXED/ ENGINEERING MODE	Prompts FIX/ENG _ and sets the Fixed/Engineering mode with the requested number of decimals
DIG	DIGITS	Sets the number of digits using X

X~~XX~~**Duplicate X**

Duplicates X into Y, Z and T

	X	Y	Z	T	L	ALPHA
Before	X					
After	X	X	X	X		

Y<>Z**Y<>Z**

Swaps the Y and the Z stack registers

	X	Y	Z	T	L	ALPHA
Before		Y	Z			
After		Z	Y			

Y<>T**Y<>T**

Swaps the Y and the T stack registers

	X	Y	Z	T	L	ALPHA
Before		Y		T		
After		T		Y		

Z<>T**Z<>T**

Swaps the Z and the T stack registers

	X	Y	Z	T	L	ALPHA
Before			Z	T		
After			T	Z		

STC**STACK CLEAR**

Clears all stack registers

	X	Y	Z	T	L	ALPHA
Before	X	Y	Z	T	L	
After	0	0	0	0	0	

STS**STACK STORE**

Stores all stack registers in registers 00-04

	X	Y	Z	T	L	ALPHA
Before	X	Y	Z	T	L	
After	R03= X	R02= Y	R01= Z	R00= T	R04= L	

STR**STACK RECALL**

Recalls registers 00-04 to stack

	R03	R02	R01	R00	R04	ALPHA
Before	R03	R02	R01	R00	R04	
After	X= R03	Y= R02	Z= R01	T= R00	L= R04	

STT**STACK TOGGLE**

Reverses the stack registers

	R03	R02	R01	R00	R04	ALPHA
Before	X	Y	Z	T	L	
After	T	Z	Y	X	L	

STV**STACK VIEW**

Displays the contents of the all the stack registers and pauses between each register

STX**STACK EXCHANGE**

Swaps all stack registers with registers 00-04.

X↔R03, Y↔R02, Z↔R01, T↔R00, L↔R04

X>=Y?	X GREATER THAN OR EQUAL TO Y	Tests if X is greater than or equal to Y	*
X>=0?	X GREATER THAN OR EQUAL TO 0	Tests if X is greater than or equal to zero	*
X=Z?	X EQUAL TO Z	Tests if X is equal to Z	*
X=T?	X EQUAL TO T	Tests if X is equal to T	*
X=1?	X EQUAL TO 1	Tests if X is equal to 1	*
Y=Z?	Y EQUAL TO Z	Tests if Y is equal to Z	*
Y=T?	Y EQUAL TO T	Tests if Y is equal to T	*
Z=T?	Z EQUAL TO T	Tests if Z is equal to T	*
X>Z?	X GREATER THAN Z	Tests if X is greater than Z	*
X<Z?	X LESS THAN Z	Tests if X is less than Z	*
X>T?	X GREATER THAN T	Tests if X is greater than T	*
X<T?	X LESS THAN T	Tests if X is less than T	*
X>1?	X GREATER THAN 1	Tests if X is greater than 1	*
X<1?	X LESS THAN 1	Tests if X is less than 1	*
		* Skips the next program line if test is false	
X=YZ?	X EQUAL TO Y OR Z	Tests if X is equal to Y or Z Skips one program line if first test (X=Y) is false. Skips two program lines if second test (X=Z) is false	
X=YZT?	X EQUAL TO Y, Z OR T	Tests if X is equal to Y,Z or T Skips one program line if first test (X=Y) is false. Skips two program lines if second test (X=Z) is false. Skips three program lines if third test (X=T) is false	

-PK1b TONE		Header ♦ Tone functions
BEEP MOD	BEEP MODE	Toggles ON/OFF a mode where HP41 beeps at any keypress
BUZZ MOD	BUZZ MODE	Toggles ON/OFF a mode where HP41 makes a 'buzz' at any keypress. <i>Only works on a real HP41</i>
TONEX	TONE X	Plays a tone determined by the value in X X<1000
TONEXY	TONE X AND Y	Plays a tone by the values in X and Y X<1000 and Y<1000. <i>Somehow Y determines the length and X determines the frequency of the tone ...</i>

CONSTAN

CONSTANTS

Constants to X *



Brings up the next menu



Aborts the menus



Puts the requested constant into X

MN MP ME e U
USER

$1,674920110 \cdot 10^{-27}$ kg	Mass of a neutron
$1,672623100 \cdot 10^{-27}$ kg	Mass of a proton
$9,109389700 \cdot 10^{-31}$ kg	Mass of an electron
$1,602177330 \cdot 10^{-19}$ C	elementary charge
$1,660570000 \cdot 10^{-27}$ kg	atomic mass unit

V0 R NO H LO
USER

22,41410000	dm ³ /mol	molar volume at 0°C
8,314510000	J/(K·mol)	gas constant
6,022136700 · 10 ²³	mol ⁻¹	Avogadro constant
6,626075500 · 10 ⁻³⁴	J·s	Planck constant
1.239,842500	nm	photon (1eV) wavelength

F G G.
USER

96.485,30900	C/mol	Faraday constant
6,672590000 · 10 ⁻¹¹	m ³ /(kg·s ²)	gravitational constant
9,806650000	m/s ²	standard gravitational acceleration

LY PS c RU
USER

9,460528405 · 10 ¹⁵	m	light-year
3,085678290 · 10 ¹⁶	m	parsec
299.792.458,0	m/s	speed of light
1,495979000 · 10 ¹¹	m	astronomical unit

* In 1990 when this program was written, these values were the accepted values

PERIOD

PERIODIC TABLE

Gets the chemical symbol of an element in the Periodic Table *

Prompts PERIOD ____ Input element number

Puts the chemical symbol into X

Input >110 gives NONEXISTENT

* In 1990 when this program was written, these were the elements known, and the accepted symbols

Hydrogen	1	H	Helium	2	He
lithium	3	Li	beryllium	4	Be
boron	5	B	carbon	6	C
nitrogen	7	N	nitrogen	8	O
oxygen	9	F	fluorine	10	Ne
neon	10	Ne	neon	11	Na
sodium	11	Na	magnesium	12	Mg
aluminum	13	Al	silicon	14	Si
silicon	14	Si	phosphorus	15	P
phosphorus	15	P	sulfur	16	S
sulfur	16	S	chlorine	17	Cl
chlorine	17	Cl	arogen	18	Ar
argon	18	Ar	potassium	19	K
potassium	19	K	calcium	20	Ca
calcium	20	Ca	scandium	21	Sc
scandium	21	Sc	titanium	22	Ti
titanium	22	Ti	vandium	23	V
vandium	23	V	chromium	24	Cr
chromium	24	Cr	manganese	25	Mn
manganese	25	Mn	iron	26	Fe
iron	26	Fe	cobalt	27	Co
cobalt	27	Co	nickel	28	Ni
nickel	28	Ni	copper	29	Cu
copper	29	Cu	zinc	30	Zn
zinc	30	Zn	gallium	31	Ga
gallium	31	Ga	germanium	32	Ge
germanium	32	Ge	arsenic	33	As
arsenic	33	As	selenium	34	Se
selenium	34	Se	broline	35	Br
broline	35	Br	krayon	36	Kr
krayon	36	Kr	rubidium	37	Rb
rubidium	37	Rb	strontium	38	Sr
strontium	38	Sr	yttrium	39	Y
yttrium	39	Y	zirconium	40	Zr
zirconium	40	Zr	niobium	41	Nb
niobium	41	Nb	ruthenium	42	Ru
ruthenium	42	Ru	rhodium	43	Rh
rhodium	43	Rh	osmium	44	Os
osmium	44	Os	iridium	45	Ir
iridium	45	Ir	platinum	46	Pt
platinum	46	Pt	osmotic	47	Ag
osmotic	47	Ag	rhodium	48	Rh
rhodium	48	Rh	osmium	49	Os
osmium	49	Os	iridium	50	Ir
iridium	50	Ir	platinum	51	Pt
platinum	51	Pt	osmotic	52	Ag
osmotic	52	Ag	rhodium	53	Rh
rhodium	53	Rh	osmium	54	Os
osmium	54	Os	iridium	55	Ir
iridium	55	Ir	platinum	56	Pt
platinum	56	Pt	osmotic	57	Ag
osmotic	57	Ag	rhodium	58	Rh
rhodium	58	Rh	osmium	59	Os
osmium	59	Os	iridium	60	Ir
iridium	60	Ir	platinum	61	Pt
platinum	61	Pt	osmotic	62	Ag
osmotic	62	Ag	rhodium	63	Rh
rhodium	63	Rh	osmium	64	Os
osmium	64	Os	iridium	65	Ir
iridium	65	Ir	platinum	66	Pt
platinum	66	Pt	osmotic	67	Ag
osmotic	67	Ag	rhodium	68	Rh
rhodium	68	Rh	osmium	69	Os
osmium	69	Os	iridium	70	Ir
iridium	70	Ir	platinum	71	Pt
platinum	71	Pt	osmotic	72	Ag
osmotic	72	Ag	rhodium	73	Rh
rhodium	73	Rh	osmium	74	Os
osmium	74	Os	iridium	75	Ir
iridium	75	Ir	platinum	76	Pt
platinum	76	Pt	osmotic	77	Ag
osmotic	77	Ag	rhodium	78	Rh
rhodium	78	Rh	osmium	79	Os
osmium	79	Os	iridium	80	Ir
iridium	80	Ir	platinum	81	Pt
platinum	81	Pt	osmotic	82	Ag
osmotic	82	Ag	rhodium	83	Rh
rhodium	83	Rh	osmium	84	Os
osmium	84	Os	iridium	85	Ir
iridium	85	Ir	platinum	86	Pt
platinum	86	Pt	osmotic	87	Ag
osmotic	87	Ag	rhodium	88	Rh
rhodium	88	Rh	osmium	89	Os
osmium	89	Os	iridium	90	Ir
iridium	90	Ir	platinum	91	Pt
platinum	91	Pt	osmotic	92	Ag
osmotic	92	Ag	rhodium	93	Rh
rhodium	93	Rh	osmium	94	Os
osmium	94	Os	iridium	95	Ir
iridium	95	Ir	platinum	96	Pt
platinum	96	Pt	osmotic	97	Ag
osmotic	97	Ag	rhodium	98	Rh
rhodium	98	Rh	osmium	99	Os
osmium	99	Os	iridium	100	Ir
iridium	100	Ir	platinum	101	Pt
platinum	101	Pt	osmotic	102	Ag
osmotic	102	Ag	rhodium	103	Rh
rhodium	103	Rh	osmium	104	Os
osmium	104	Os	iridium	105	Ir
iridium	105	Ir	platinum	106	Pt
platinum	106	Pt	osmotic	107	Ag
osmotic	107	Ag	rhodium	108	Rh
rhodium	108	Rh	osmium	109	Os
osmium	109	Os	iridium	110	Ir
iridium	110	Ir	platinum	111	Pt
platinum	111	Pt	osmotic	112	Ag
osmotic	112	Ag	rhodium	113	Rh
rhodium	113	Rh	osmium	114	Os
osmium	114	Os	iridium	115	Ir
iridium	115	Ir	platinum	116	Pt
platinum	116	Pt	osmotic	117	Ag
osmotic	117	Ag	rhodium	118	Rh
rhodium	118	Rh	osmium	119	Os
osmium	119	Os	iridium	120	Ir
iridium	120	Ir	platinum	121	Pt
platinum	121	Pt	osmotic	122	Ag
osmotic	122	Ag	rhodium	123	Rh
rhodium	123	Rh	osmium	124	Os
osmium	124	Os	iridium	125	Ir
iridium	125	Ir	platinum	126	Pt
platinum	126	Pt	osmotic	127	Ag
osmotic	127	Ag	rhodium	128	Rh
rhodium	128	Rh	osmium	129	Os
osmium	129	Os	iridium	130	Ir
iridium	130	Ir	platinum	131	Pt
platinum	131	Pt	osmotic	132	Ag
osmotic	132	Ag	rhodium	133	Rh
rhodium	133	Rh	osmium	134	Os
osmium	134	Os	iridium	135	Ir
iridium	135	Ir	platinum	136	Pt
platinum	136	Pt	osmotic	137	Ag
osmotic	137	Ag	rhodium	138	Rh
rhodium	138	Rh	osmium	139	Os
osmium	139	Os	iridium	140	Ir
iridium	140	Ir	platinum	141	Pt
platinum	141	Pt	osmotic	142	Ag
osmotic	142	Ag	rhodium	143	Rh
rhodium	143	Rh	osmium	144	Os
osmium	144	Os	iridium	145	Ir
iridium	145	Ir	platinum	146	Pt
platinum	146	Pt	osmotic	147	Ag
osmotic	147	Ag	rhodium	148	Rh
rhodium	148	Rh	osmium	149	Os
osmium	149	Os	iridium	150	Ir
iridium	150	Ir	platinum	151	Pt
platinum	151	Pt	osmotic	152	Ag
osmotic	152	Ag	rhodium	153	Rh
rhodium	153	Rh	osmium	154	Os
osmium	154	Os	iridium	155	Ir
iridium	155	Ir	platinum	156	Pt
platinum	156	Pt	osmotic	157	Ag
osmotic	157	Ag	rhodium	158	Rh
rhodium	158	Rh	osmium	159	Os
osmium	159	Os	iridium	160	Ir
iridium	160	Ir	platinum	161	Pt
platinum	161	Pt	osmotic	162	Ag
osmotic	162	Ag	rhodium	163	Rh
rhodium	163	Rh	osmium	164	Os
osmium	164	Os	iridium	165	Ir
iridium	165	Ir	platinum	166	Pt
platinum	166	Pt	osmotic	167	Ag
osmotic	167	Ag	rhodium	168	Rh
rhodium	168	Rh	osmium	169	Os
osmium	169	Os	iridium	170	Ir
iridium	170	Ir	platinum	171	Pt
platinum	171	Pt	osmotic	172	Ag
osmotic	172	Ag	rhodium	173	Rh
rhodium	173	Rh	osmium	174	Os
osmium	174	Os	iridium	175	Ir
iridium	175	Ir	platinum	176	Pt
platinum	176	Pt	osmotic	177	Ag
osmotic	177	Ag	rhodium	178	Rh
rhodium	178	Rh	osmium	179	Os
osmium	179	Os	iridium	180	Ir
iridium	180	Ir	platinum	181	Pt
platinum	181	Pt	osmotic	182	Ag
osmotic	182	Ag	rhodium	183	Rh
rhodium	183	Rh	osmium	184	Os
osmium	184	Os	iridium	185	Ir
iridium	185	Ir	platinum	186	Pt
platinum	186	Pt	osmotic	187	Ag
osmotic	187	Ag	rhodium	188	Rh
rhodium	188	Rh	osmium	189	Os
osmium	189	Os	iridium	190	Ir
iridium	190				

GEOPOS **GEOGRAPHICAL POSITION**

Puts latitude and longitude of 24 cities into X and Y *

Displays a sequence of cities.



Brings up the next.



Aborts the sequence.



Puts latitude into X and longitude into Y

See next page for a map with the cities

CITY	LATITUDE	LONGITUDE
FREDERICIA	55,34	-9,45
TREND	56,51	-9,13
KØBENHAVN	55,41	-12,34
AARHUS	56,08	-10,11
AALBORG	57,02	-9,54
ODENSE	55,22	-10,23
ESBJERG	55,28	-8,26
SKAGEN	57,43	-10,35
NEXØ	55,04	-15,08
GEDSER	54,35	-11,55
GODTHAAB	64,10	51,35
LONDON	51,30	0,00
PARIS	48,52	-2,20
NEW YORK	40,43	74,01
TOKYO	35,42	-139,46
MUNCHEN	48,08	-11,34
REYKJAVIK	64,09	21,51
MOSKVA	55,45	-37,35
ROM	41,54	-12,29
SINGAPORE	1,17	-103,51
SYDNEY	-33,50	-151,10
LOS ANGELES	34,00	118,10
RIO DE JAN	-23,00	43,12
HAWAII	20,30	157,00

* In 1990 when this program was written, I wasn't aware of the fact that my friend Ángel was born in Madrid, so I'm afraid it's not in there...

** Godthaab is now called Nuuk, and Godthaab is very politically incorrect!

EXP **EXPONENT**

Puts an exponent into X *

Displays a sequence of exponent names.



Brings up the next.



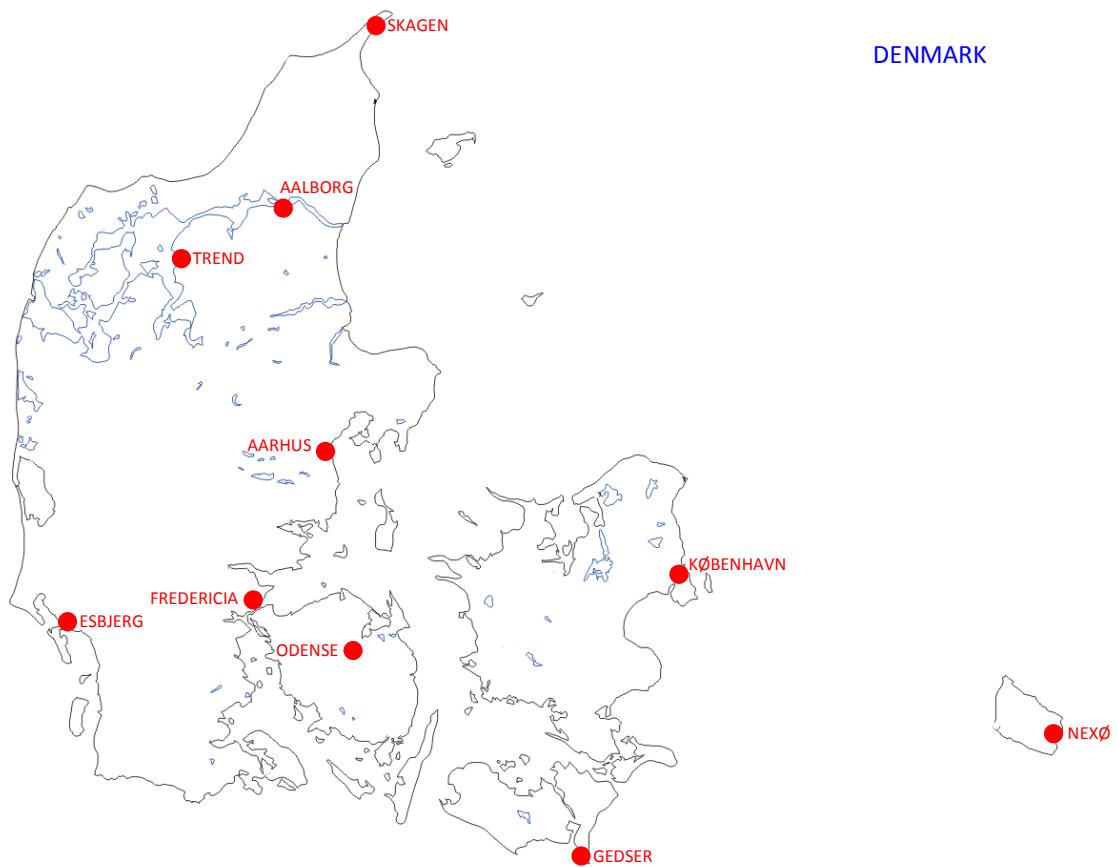
Aborts the sequence.



Puts the exponent value into X

EXPONENT	VALUE
MILLION	$1 \cdot 10^6$
MILLIARD	$1 \cdot 10^9$
BILLION	$1 \cdot 10^{12}$
BILLIARD	$1 \cdot 10^{15}$
TRILLION	$1 \cdot 10^{18}$
TRILLIARD	$1 \cdot 10^{21}$
KVARTILLION	$1 \cdot 10^{24}$

* In 1990 when this program was written, these were the accepted **danish** names and their values



PK2c

-PK2c MATH

Header • Math functions

SINH	HYPERBOLIC SINE	Calculates SINH(X)	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>sinh(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	sinh(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	sinh(X)				X																			
COSH	HYPERBOLIC COSINE	Calculates COSH(X)	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>cosh(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	cosh(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	cosh(X)				X																			
TANH	HYPERBOLIC TANGENT	Calculates TANH(X)	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>tanh(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	tanh(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	tanh(X)				X																			
ASINH	AREA HYPERBOLIC SINE	Calculates ASINH(X)	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>asinh(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	asinh(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	asinh(X)				X																			
ACOSH	AREA HYPERBOLIC COSINE	Calculates ACOSH(X)	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>acosh(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	acosh(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	acosh(X)				X																			
ATANH	AREA HYPERBOLIC TANGENT	Calculates ATANH(X)	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>atanh(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	atanh(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	atanh(X)				X																			
CBRT	CUBICROOT	Calculates $\sqrt[3]{x}$	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>cbrt(X)</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	cbrt(X)				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	cbrt(X)				X																			
X ³	CUPIC POWER	Calculates X ³	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>X³</td><td></td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	X ³				X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	X ³				X																			
BFACT	BIG FACTORIAL	Calculates X! for X>69	<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>After</td><td>Signicand (X!)</td><td>Exponent (X!)</td><td></td><td></td><td>X</td><td></td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	X						After	Signicand (X!)	Exponent (X!)			X	
	X	Y	Z	T	L	ALPHA																		
Before	X																							
After	Signicand (X!)	Exponent (X!)			X																			

BYX

BIG POWERS

Calculates Y^X for big numbers

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	Significand (y^x)	Exponent (y^x)			X	

QR

QUOTIENT AND
REMAINDER

Calculates the quotient and remainder of X and Y

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	remainder	quotient			X	

GCD

GREATEST
COMMON DIVIDER

Calculates the greatest common divider of X and Y

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	gcd(X,Y)				X	

LCM

LEAST COMMON
MULTIPLE

Calculates the least common multiple of X and Y

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	lcm(X,Y)				X	

DIVISOR

DIVISOR

Finds the smallest divider of a X

	X	Y	Z	T	L	ALPHA
Before	X					
After	divisor(X)	X			X	

PRIME?

PRIME TEST

Tests if X is a prime or not. Displays YES or NO and beeps a high or low tone.

Skips the next program line if test is false

	X	Y	Z	T	L	ALPHA
Before	X					
After	X				X	

NXTPRIM

NEXT PRIME

Finds the next prime number larger than X. Displays it and sounds a tone

	X	Y	Z	T	L	ALPHA
Before	X					
After	next prime				X	

INTY

INTEGER PART OF Y

Calculates the integer part of Y

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	X	int(Y)				

FRCY

FRACTIONAL PART
OF Y

Calculates the fractional part of Y

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	X	frc(Y)				

UNARY

UNARY

Returns:

If $X > 0$:

	X	Y	Z	T	L	ALPHA
Before	X					
After	1				X	

If $X = 0$:

Before	0					
After	0				0	

If $X < 0$:

Before	X					
After	-1				X	

VWM

VIEW MANTISSA

Views the significand part of X

	X	Y	Z	T	L	ALPHA
Before	X					
After	X					

 Σ

SUM OF DIGITS

Calculates the sum of the digits in the mantissa

	X	Y	Z	T	L	ALPHA
Before	X					
After	$\Sigma(\text{digits})$				X	

 $\Sigma 1$

SUM OF NUMBERS

Calculates the sum of the numbers from 1 to X (X must be integer !)

	X	Y	Z	T	L	ALPHA
Before	X					
After	$\Sigma(X)$				X	

 $\Sigma 2$

SUM OF SQUARES

Calculates the sum of the numbers from 1 to X squared (X must be integer !)

	X	Y	Z	T	L	ALPHA
Before	X					
After	$\Sigma(X^2)$				X	

 $\Sigma 3$

SUM OF CUBES

Calculates the sum of the numbers from 1 to X cubed (X must be integer !)

	X	Y	Z	T	L	ALPHA
Before	X					
After	$\Sigma(X^3)$				X	

K

KOMBINATIONER

Calculates $C(Y,X)$ (Y must be larger than X and less than 100)

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	$C(Y,X)$				X	

P

PERMUTATIONER

Calculates $P(Y,X)$ (Y must be larger than X and less than 100)

	X	Y	Z	T	L	ALPHA
Before	X	Y				
After	$P(Y,X)$				X	

S-R	SPHERICAL TO RECTANGULAR	Calculates spherical coordinates to rectangular coordinates				
			X	Y	Z	T
	Before	R	θ	ϕ		L
	After	X	Y	Z		R
R-S	RECTANGULAR TO SPHERICAL	Calculates rectangular coordinates to spherical coordinates				
			X	Y	Z	T
	Before	X	Y	Z		L
	After	R	θ	ϕ		X
HERON	HERON'S FORMULA	Calculates the area of a triangle given the sides a,b and c				
			X	Y	Z	T
	Before	5	12	13		L
	After	30				5
X%Y	X percentage of Y	Calculates the percentage of X according to Y				
			X	Y	Z	T
	Before	11	25			L
	After	44				11
PYTH	PYTHAGORAS	Calculates and displays/prints triplets of integer numbers that fit into Pythagoras equation Prompts for P? which is a starting value, used by the formula involved. If you want small numbers choose a small P, if you want larger numbers choose a larger P. A value of 3 gives 5 12 13 and 5 8 17 as the first triplets				
dB+	DECIBEL ADDITION	Adds soundlevels in X and Y				
			X	Y	Z	T
	Before	10	20			L
	After	20,414				10
dB-	DECIBEL SUBTRACTION	Subtracts soundlevels in X and Y				
			X	Y	Z	T
	Before	10	20,414			L
	After	20				10

KOMB

KOMBINATORIK

Calculates some combinatorics * uses CCD & PPC

Displays

n and p must be <100

LBL A

Ordnet sampling uden replacement

n p

Input 10 ↗ 5 gives 30240

LBL B

Ordnet sampling med replacement

n p

Input 10 ↗ 5 gives 100000

LBL C

Uordnet sampling uden replacement

n p

Input 10 ↗ 5 gives 252

LBL D

Uordnet sampling med replacement

n p

Input 10 ↗ 5 gives 2002

LBL E

Hypergeometric ventetidsfordeling

Prompts N? Input number of balls

Prompts R? Input number of red balls

Prompts X? Input number of failures

Displays V(X)= probability of X failures before a red ball is drawn

Input of 10, 5 and 2 gives V(2)=0.1389

i.e. with 5 red balls out of 10 the probability of 2 failures before a red ball is drawn is 13,89 %

LBL F

Uordnet gruppering

Prompts ANTAL? Input number of balls

Prompts ANTAL GR.SET? Input number of groups with equal size

Prompts 1.GR SIZE↗NMB Input size and number of groups with that size

Prompts 2.GR SIZE↗NMB Input size and number of groups with that size

Displays number of permutations

Input of 10, 2, 2 ↗ 2 and 3 ↗ 2 gives 6300

i.e. ...

LBL G

Ordnet gruppering

Prompts ANTAL? Input number of balls

Prompts ANTAL GR.? Input number of groups

Prompts 1? Input 1st group size

Prompts 2? Input 2nd group size etc.

Displays number of permutations

Input of 11, 4, 1, 4, 4 and 2 gives 34650

i.e. there are 34650 distinct permutations of the word MISSISSIPPI

LBL H

Hypergeometric distribution

Prompts N? Input number of balls

Prompts U? Input number of balls in sample

Prompts R? Input number of red balls

Prompts X? Input number of red balls in sample

Displays P(X)= probability of X red balls in the sample

Input of 40, 10, 5 and 1 gives P(1)=0,4165

i.e. with 5 red balls out of 40 and a sample of 10 the probability of getting 1 red is 41,65 %

LBL I

Binomial distribution

BINOM
USER

Prompts **N?** Input number of repeats
 Prompts **%?** Input probability of primary event
 Prompts **X?** Input number of primary events

Displays $b(X) =$ probability of X primary eventsInput of 10, 0.5 and 2 gives $b(10,2) = 0,0439$

i.e. with a probability of 50% the chance of 2 primary events out of 10 is 4,39%

LBL d

N-stikprøve uden replacement

N-STIKPROVE
USER

Prompts **I?** Input number of colors
 Prompts **M1?** Input number of balls in 1st color
 Prompts **M2?** Input number of balls in 2nd color etc.
 Prompts **X1?** Input number of balls of 1st color in sample
 Prompts **X2?** Input number of balls in 2nd color in sample
 Displays $P(X_1, X_2, \dots, X_i) =$ probability of X_1, X_2, \dots, X_i balls in sample

Input of 3, 3, 4, 5 and 1, 2, 3 gives $P(1,2,3) = 0,1948$

*

LBL e

Multinomial distribution

MULTINOM
USER

Prompts **I?** Input number of events
 Prompts **P1?** Input probability of 1st event
 Prompts **P2?** Input probability of 2nd event etc.
 Prompts **X1?** Input number of primary events
 Prompts **X2?** Input number of secondary events etc
 Displays $P(X_1, X_2, \dots, X_i) =$ probability of X_1, X_2, \dots, X_i events

Input of 3, 0.25, 0.30, 0.45 and 2, 3, 4 gives $P(2,3,4) = 0,0872$

* Prompts for X1 again, and program is in the INP? routine

This means LBL J isn't available

LBL J

Menu

2POLY

2.GRADS

POLYNOMIUM

Tools for Quadratic Polynomials

Displays		
LBL A	Find the roots	
LBL B	Prompts TOPP. ?X?Y	Input Turning Point (Vertex) of Poly
	Prompts A?	Input coefficient of X^2
	Displays the Equation of the polynomium	
LBL C	Prompts X?	Evaluates the poly with the given X
LBL E	Prompts a?b?c	Input coefficients of the polynomium
LBL F	Given 3 points it calculates the equation	
LBL G	Displays the derivative of the poly	
LBL J	Displays menu	

An example:

LBL E	Prompts a?b?c	Input 1 ↗ 2 ↗ -8
LBL A	See	$TOPP. = (-1.00X - 9.00)$ $X1 = -4.00$ $X2 = 2.00$ $D = 36.00$
LBL C	Prompts X?	Input 10
	see	$X, Y = (10.00X + 12.00)$
LBL G	See	$Y = 2.00X + 2.00$

Another example:

LBL B	Prompts TOPP. ?X?Y	Input 1 ↗ -8
	Prompts A?	Input 2
	See	$Y = +2.00X^2 - 4.00X - 6.00$

A third example:

LBL E	Prompts a?b?c	Input 1 ↗ 2 ↗ 8
LBL A	See	$TOPP. = (-1.00X^2 + 7.00)$ $D = -28.00$

A fourth example:

LBL F	Prompts X1?Y1?	Input 1st point 1 ↗ -1
	Prompts X2?Y2?	Input 2nd point 2 ↗ 0
	Prompts X3?Y3?	Input 3rd point 3 ↗ 3
	See	$Y = +1.0X^2 - 2.0X + 3.0E-10$

POLY

POLYNOMIALS

Multiples or divides two polynomials * uses CCD and ADVANTAGE

Displays

The screen shows a menu with four items: IN, *, /, and R. Below these, there is a small 'USER' label.

LBL A Input polynomials

Prompts DEG.1=? Input degree of 1st poly

Prompts DEG.2=? Input degree of 2nd poly

Displays INPUT 1.POLY

Prompts aN=? Input coefficient of x^n continues with x^{n-1} etc.

Displays INPUT 2.POLY

Prompts bN=? Input coefficient of x^n continues with x^{n-1} etc.

Displays menu

LBL a Input new 2nd polynomial

LBL B Multiplies the two polynomials

LBL C Divides the two polynomials

LBL c ???

LBL D Calls line 28 in the PLY program from ADVANTAGE

The screen shows a menu with three items: F X, R T, and NEW. Below these, there is a small 'USER' label.

LBL E View the results

LBL e View rest polynomial when dividing

LBL J Displays menu

An example:

Multiply $2 \cdot x^2 + 3 \cdot x - 4$ and $3 \cdot x + 5$

LBL A

Prompts DEG.1=? Input 2

Prompts DEG.2=? Input 1

Prompts a2=? Input 2

Prompts a1=? Input 3

Prompts a0=? Input -4

Prompts b1=? Input 3

Prompts b0=? Input 5

LBL B multiply

LBL E see $a_3 = 6,00$
 $a_2 = 19,00$
 $a_1 = 3,00$
 $a_0 = -20,00$

i.e. result is $6 \cdot x^3 + 19 \cdot x^2 + 3 \cdot x - 20$

Another example:

Divide $6 \cdot x^3 + 19 \cdot x^2 + 3 \cdot x - 20$ and $2 \cdot x^2 + 5 \cdot x + 2$

LBL a 1st poly is already there so we only need to input 2nd poly

Prompts DEG.2=? Input 2

Prompts b2=? Input 2

Prompts b1=? Input 5

Prompts b0=? Input 2

LBL C divide

LBL E see $a_1 = 3,00$
 $a_0 = 2,00$

i.e. quotient is $3 \cdot x + 2$

LBL e see $a_3 = 0,00$
 $a_2 = 0,00$
 $a_1 = -13,00$
 $a_0 = -24,00$

and remainder is $-13 \cdot x - 24$

Calculates and prints variables on a triangle * 3 inputs are required

Prompts $\angle A$	Input angle A or press	R/S	if unknown
Prompts $\angle B$	Input angle B or press	R/S	if unknown
Prompts $\angle C$	Input angle C or press	R/S	if unknown
Prompts a	Input length of a or press	R/S	if unknown
Prompts b	Input length of b or press	R/S	if unknown
Prompts c	Input length of c or press	R/S	if unknown

With input $\angle A=45$, $\angle C=30$ and $b=6$ the program views/prints the following:

$\angle A = 45,00$
 $\angle B = 105,00$
 $\angle C = 30,00$
 $a = 4,39$
 $b = 6,00$
 $c = 3,11$

And then prompts CON? _ for additional information press J, if not press N.

T = 6,59	area
H _a = 3,00	h_a length of altitudes
H _b = 2,20	h_b
H _c = 4,24	h_c
M _a = 4,24	m_a length of medians
M _b = 2,34	m_b
M _c = 5,02	m_c
V _a = 3,78	v_a length of angle bisectors
V _b = 2,22	v_b
V _c = 4,90	v_c
R.O = 3,11	radius of circumcircle
R.I = 0,98	radius of incircle
R.a = 2,80	radius of excircle _a
R.b = 8,80	radius of excircle _b
R.c = 1,81	radius of excircle _c
A = (0;0)	A vertices
B = (2,20×2,20)	B
C = (6,00×0,00)	C
H' SK = (2,20×3,80)	intersection of altitudes
M' SK = (2,73×0,73)	intersection of medians = centroid
V' SK = (2,36×0,98)	intersection of bisectors
=C-R.I	= incenter
N' SK = (3,00×-0,80)	intersection of perpendicular bisectors
=C-R.O	= circumcenter
C-Ra = (6,75×2,80)	center of excircle _a
C-Rb = (3,64×-8,80)	center of excircle _b
C-Rc = (-0,75×1,81)	center of excircle _c

PGA

POLYGON AREA

Calculates the area of a polygon

Displays

$R =$
USER

* Turn on USER !

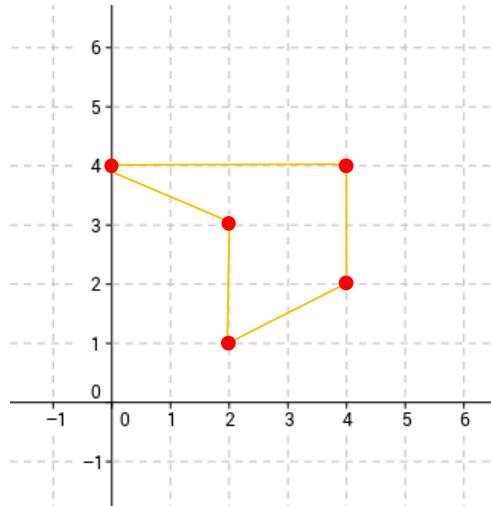
Input $X \nearrow Y$ of first point press \sqrt{x}

Repeat this for all points of the Polygon.

The input has to be clockwise or counterclockwise!

Press $\Sigma +$ Displays $A =$ area of polygon

Input of $2 \nearrow 1, 4 \nearrow 2, 4 \nearrow 4, 4 \nearrow 0$ and $2 \nearrow 3$ gives $A=6.00$



CIRKEL

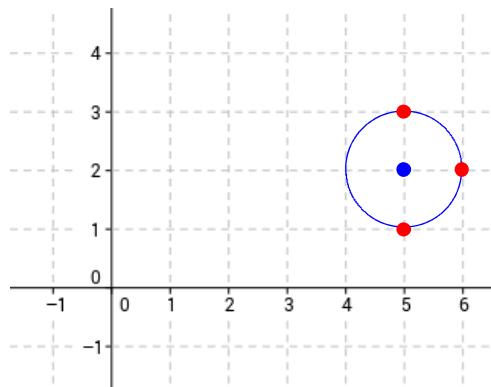
CIRCLE FROM 3
POINTS

Calculates the radius and center of a circle, given 3 points

Prompts for three points: $X1 \nearrow Y1$, $X2 \nearrow Y2$ and $X3 \nearrow Y3$

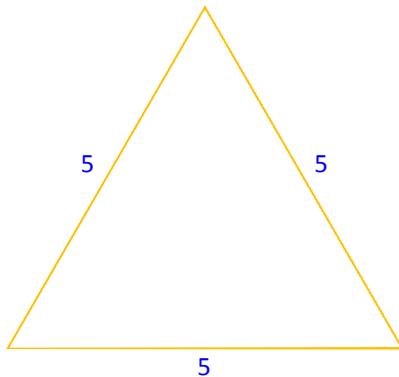
Displays $R =$ radius and $C = (C1x, C2)$

Input of $5 \nearrow 1, 6 \nearrow 2$ and $5 \nearrow 3$ gives $R=1.00$ and $C=(5.00, 2.00)$



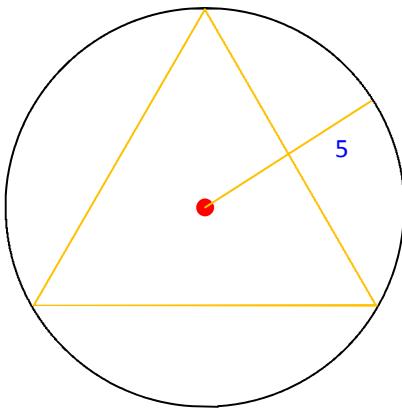
PG1

AREA OF POLYGON Calculates the area of a regular polygon
1 Input number of sides
Input of 3 ↗ 5 gives 10.83



PG2

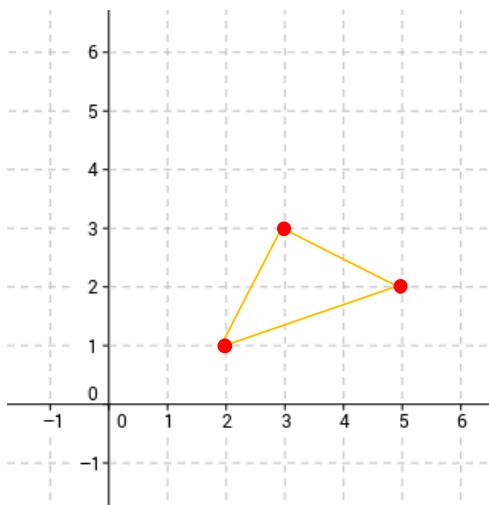
AREA OF POLYGON Calculates the area of a regular polygon
2 Input number of sides
Input radius of circumscribed circle XEQ PG2
Input of 3 ↗ 5 gives 32.48

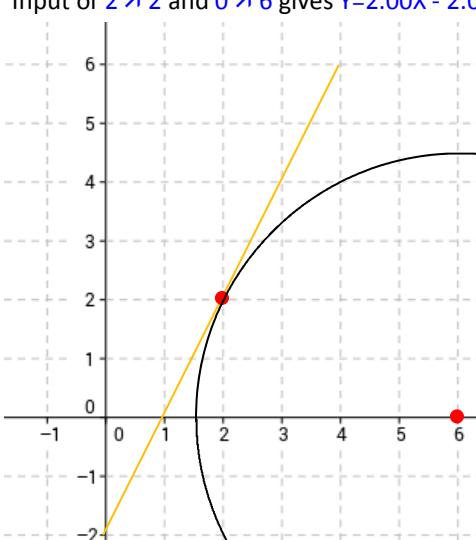
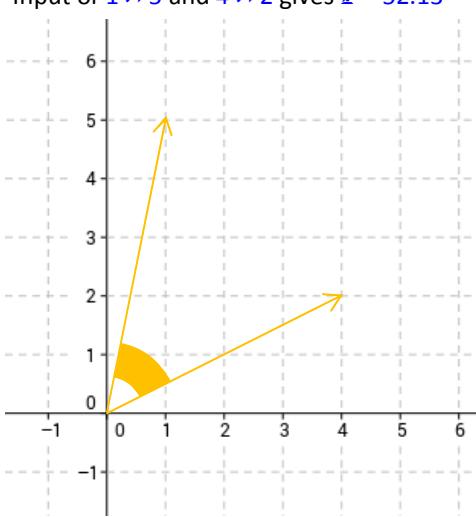


3AREAL

3KANT AREAL Calculates the area of a triangle
Prompts A1↗A2 Input 1st point
Prompts B1↗B2 Input 2nd point
Prompts C1↗C2 Input 3rd point
Displays A= area

Input of 2 ↗ 1 , 5 ↗ 2 and 3 ↗ 3 gives A=2.50



CIRKTAN	TANGENT TO A CIRCLE	Calculates and displays the tangent line to a circle Prompts $P_1 \triangleright P_2$ Input touching point $Y \not\propto X !!$ Prompts $C_1 \triangleright C_2$ Input center of circle $Y \not\propto X !!$ Displays $Y = a \cdot X + b$ the tangent line
		Input of $2 \triangleright 2$ and $0 \triangleright 6$ gives $Y = 2.00X - 2.00$
		
\angle	ANGLE	Calculates the angle between two vectors Prompts $a_1 \triangleright a_2$ Input 1st vector Prompts $b_1 \triangleright b_2$ Input 2nd vector Displays $\angle = \text{angle}$
		Input of $1 \triangleright 5$ and $4 \triangleright 2$ gives $\angle = 52.13$
		
LR	LINEAR REGRESSION	Calculates and displays Correlation of a set of data-points Clear stat memory $CL\sigma$ Input a set of corresponding X and Y values $\Sigma+$ XEQ LR Displays $R = r$ the correlation coefficient Displays $a = a$ and $b = b$ $Y = a \cdot X + b$ which is the best fitting line through your data-points

AFSTAND

AFSTAND

Calculates and displays the distance between a line and a point

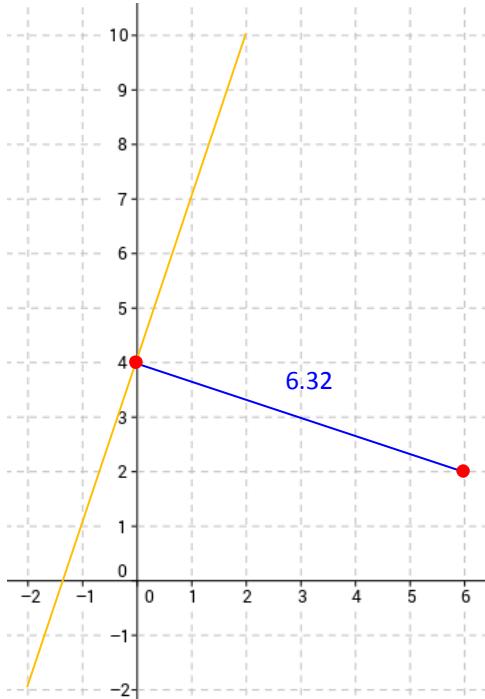
Displays $aX+bY+c=0$

Prompts $a \triangleright b \triangleright c$ Input coefficients

Prompts $X \triangleright Y$ Input coordinates of point

Displays DIST= distance

Input of $3 \triangleright -1 \triangleright 4$ and $6 \triangleright 2$ gives DIST=6.32



LINIE

LINIE GENNEM TO
PUNKTER

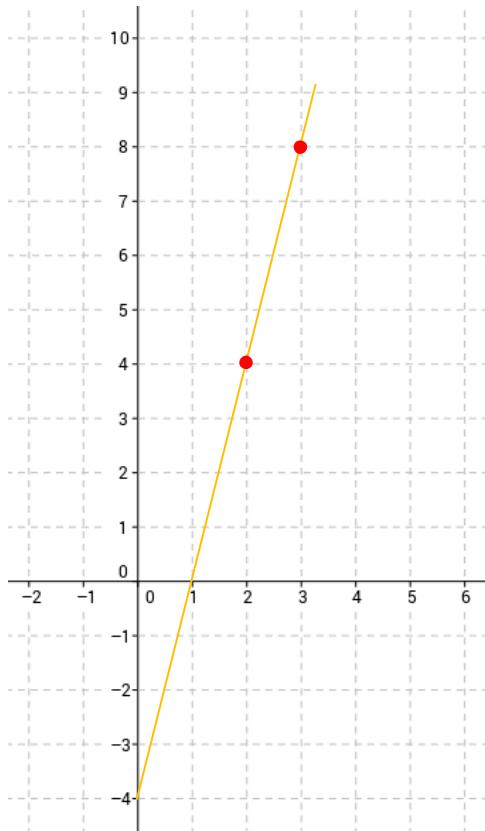
Calculates and displays the equation of a straight line through two points

Prompts $X1 \triangleright Y1$ Input coordinates of the first point

Prompts $X2 \triangleright Y2$ Input coordinates of the second point

Displays $Y = ax + b$ a is in X and b is in Y

Input of $2 \triangleright 4$ and $3 \triangleright 8$ gives $Y = 4.00X - 4.00$



KEGLESN

KEGLESNIT

Calculates and prints Conic Sections

$$a \cdot x^2 + b \cdot xy + c \cdot y^2 + d \cdot x + e \cdot y + f = 0$$

Prompts $a \cdot x^2$ Input a

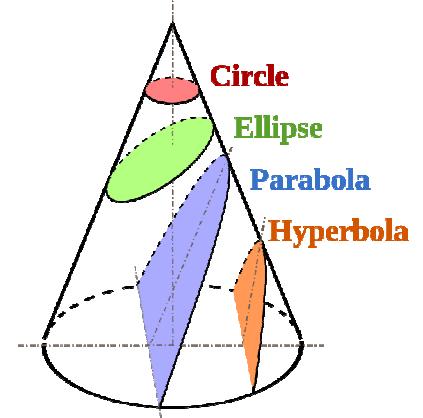
Prompts $b \cdot xy$ Input b

Prompts $c \cdot y^2$ Input c

Prompts $d \cdot x$ Input d

Prompts $e \cdot y$ Input e

Prompts f Input f



Input $1, 0, 1, 0, 0, 0$ views/prints the following:

$\angle = 0,00$
PUNKT
 $(X,Y) = (0,00 \times 0,00)$

Angle = 0
Point
(x,y) = (0,0)

Input $1, 0, 1, 0, 0, -1$ views/prints the following:

$\angle = 0,00$
CIRKEL
 $R = 1,00$
CENTR. = $(0,00 \times 0,00)$

Angle = 0
Circle
Radius = 1.00
Center = (0,0)

Input $1, 4, 1, 0, 0, 0$ views/prints the following:

$\angle = 45,00$
 $X2 = 3,00$
 $Y2 = -1,00$
 $X = 0,00$
 $Y = 0,00$
= 0,00
LINIER
 $Y = -1,73X$
OPR. SYS.
 $Y = -0,27X$
 $Y = 1,73X$
OPR. SYS.
 $Y = -3,73X$

Input $1, 4, 6, -4, 5, -9$ views/prints the following:

$\angle = -19,33$
 $X2 = 0,30$
 $Y2 = 6,70$
 $X = -5,43$
 $Y = 3,39$
= -9,00
ELLIPSE
 $a = 10,69$ $b = 2,26$
 $e = 0,98$
CENTR. = $(9,10 \times -0,25)$
O.S. = $(8,50 \times -3,25)$
 $F1 = (19,55 \times -0,25)$
 $O.S. = (18,36 \times -6,71)$
 $F2 = (-1,36 \times -0,25)$
O.S. = $(-1,36 \times 0,21)$
LEDELINIER:
 $X = 20,04$
OPR. SYS.
 $Y = -0,35X + 21,23$
 $X = -1,84$
OPR. SYS.
 $Y = -0,35X - 1,95$

(*)()	MULTIPLY PARENTHESES	Multiplication of two parentheses uses XF, ADVANTAGE, PK1, PK3
	<i>An example:</i>	$(5a+7b) \cdot (9c+11d+13e)$
	Displays 1.PARANTES	
	Prompts ANTAL LED?	Input number of parts in 1st parentheses 2
	Prompts TEXT: _	Input 1st text A
	Prompts A ?	Input coefficient of A 5
	Prompts TEXT: _	Input 2nd text B
	Prompts B ?	Input coefficient of B 7
	Displays 2.PARANTES	
	Prompts ANTAL LED?	Input number of parts in 2nd parentheses 3
	Prompts TEXT: _	Input 1st text C
	Prompts C ?	Input coefficient of C 9
	Prompts TEXT: _	Input 2nd text D
	Prompts D ?	Input coefficient of D 11
	Prompts TEXT: _	Input 3rd text E
	Prompts E ?	Input coefficient of E 13
	See	AC +45 AD +55 AE +65 BC +63 BD +77 BE +91 i.e. 45ac+55ad+65ae+63bc+77bd+91be
	Prompts CONT' ? N:C:S	N is for new calculation C is for continued calculation with the above S is for STOP
LGI	LAGRANGE INTERPOLATION	Performs LaGrange Interpolation on a set of data * uses CX Extended
	Prompts PKT?	Input number of data points
	Prompts X?Y (1)	Input 1st point
	Prompts X?Y (2)	Input 2nd point, and then continue with all points
	Prompts X?	Input X to be evaluated
	<i>An example:</i>	
	Input 7 points: (-3,-15), (-2,1), (-1,5), (0,3), (1,1), (2,5) and (3,21)	
	Input X: -1.65 and see 3.4579	
	Input X: 0.2 and see 2.4080	

PTEGN

PERSPEKTIV

TEGNING

Calculates coordinates for plotting 3-D objects in 2-D * uses HP-IL (for saving/reading data only)

Displays

P I P X Y E I E
USER

LBL A

Input P-data:

Prompts X-TRAN. = Input x transition

Prompts Y-TRAN. = Input y transition

Prompts Z-TRAN. = Input z transition

Prompts P.DIST. = Input distance

Prompts X-ROT. = Input x rotation

Prompts Y-ROT. = Input y rotation

Prompts Z-ROT. = Input z rotation

LBL a

View P-data:

P-DATA :

X-TRAN. = 1.0

Y-TRAN. = 2.0

Z-TRAN. = 3.0

P.DIST. = 5.0

X-ROT. = 0.0

Y-ROT. = 0.0

Z-ROT. = 0.0

LBL B

Save P-data:

Prompts FIL: _ Input filename

Creates data-file on HP-IL Drive and writes P-data

LBL b

Read P-data:

Prompts FIL: _ Input filename

Reads P-data from HP-IL Drive

LBL C

Calculates the coordinates for plotting on millimeter-paper

LBL c

Views/prints the coordinates:

A

(1.0 3.0 2.0)

X: 2.0 Y: 5.0

B

(5.0 3.0 2.0)

X: 6.0 Y: 5.0

C

(1.0 6.0 2.0)

X: 1.3 Y: 3.1

D

(5.0 6.0 2.0)

X: 3.8 Y: 3.1

E

(1.0 3.0 9.0)

X: 2.0 Y: 12.0

F

(5.0 3.0 9.0)

X: 6.0 Y: 12.0

G

(1.0 6.0 9.0)

X: 1.3 Y: 7.5

H

(5.0 6.0 9.0)

X: 3.7 Y: 7.5

LBL D

Input C-data

Prompts X Y Z Input coordinates of points

End with "empty" R/S to update counter of points !

LBL d

View C-data:

C-DATA :

(1.0 3.0 2.0)

(5.0 3.0 2.0)

(1.0 6.0 2.0)

(5.0 6.0 2.0)

(1.0 3.0 9.0)

(5.0 3.0 9.0)

(1.0 6.0 9.0)

(5.0 6.0 9.0)

LBL E

Save C-data:

Prompts FIL: _

Input filename

Creates data-file on HP-IL Drive and writes C-data

LBL e

Read C-data:

Prompts FIL: _

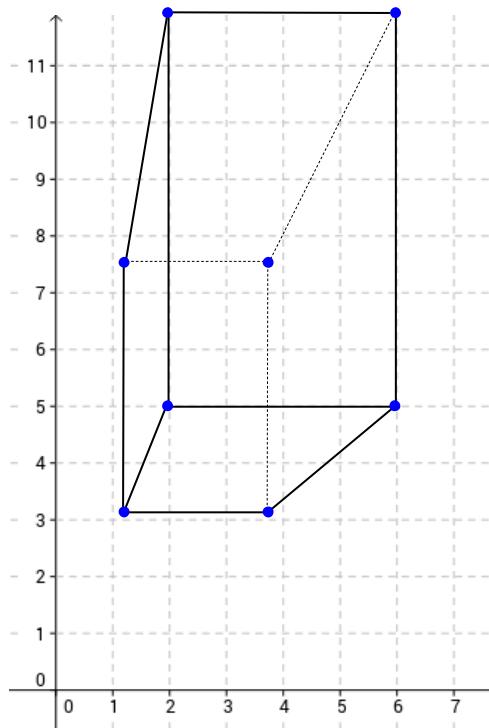
Input filename

Reads C-data from HP-IL Drive

LBL J

Displays menu

An example using the data on page 34:



CP2

**COMPLEX
QUADRATIC
POLYNOMIAL**

Solves a complex quadratic polynomial * uses ADVANTAGE

Prompts **X+YI ? Z \geq 2** Input coefficient of z^2 Prompts **X+YI ? Z \geq 1** Input coefficient of z^1 Prompts **X+YI ? Z \geq 0** Input coefficient of z^0 *An example:*

$$(1+i) \cdot z^2 + 5i \cdot z = 0$$

Input **1 \geq 1, 5 \geq 0 and 0 \geq 0**

See U= 0.0000 i.e $z_1 = 0$
 V= 0.0000
 U= -2.5000
 V= -2.5000 $z_2 = -2.5 - 2.5i$

Another example:

$$2i \cdot z^2 + (-2 - 3i) \cdot z + (-11 - 10i) = 0$$

Input **2 \geq 0, -3 \geq -2 and -10 \geq -11**

See U= 3.3495 i.e $z_1 = 3.35 - 1.70i$
 V= -1.7021
 U= -1.8495
 V= 0.7021 $z_2 = -1.85 + 0.70i$

CHT

**COMPLEX
HYPERBOLICS AND
TRIGONOMETRY**

Inverse Complex Hyperbolic and Trigonometry * uses ADVANTAGE

Displays  USERInput **X + Yi**

LBL A Inverse sine
 Input **1 \geq 1** and get **0.6662** in X and **1.0613** in Y
 i.e. $\sin^{-1}(1 + i)$ is **0.6662 + 1.0613i**
 Input **2 \geq 2** and get **0.7542** in X and **1.7343** in Y

LBL a Inverse hyperbolic sine
 Input **1 \geq 1** and get **1.6013** in X and **0.6662** in Y
 Input **2 \geq 2** and get **1.7343** in X and **0.7542** in Y

LBL B Inverse cosine
 Input **1 \geq 1** and get **0.9046** in X and **-1.0613** in Y
 Input **2 \geq 2** and get **0.8165** in X and **-1.7343** in Y

LBL b Inverse hyperbolic cosine
 Input **1 \geq 1** and get **1.6013** in X and **0.9046** in Y
 Input **2 \geq 2** and get **1.7343** in X and **0.8165** in Y

LBL C Inverse tangent
 Input **1 \geq 1** and get **1.0172** in X and **0.4024** in Y
 Input **2 \geq 2** and get **1.3112** in X and **0.2389** in Y

LBL c Inverse hyperbolic tangent
 Input **1 \geq 1** and get **0.4024** in X and **1.0172** in Y
 Input **2 \geq 2** and get **0.2389** in X and **1.3112** in Y

LBL J Displays menu

aX+bY=c	aX+bY=c	Calculates integer solutions to aX+bY=c Prompts a? Input a Prompts b? Input b Prompts c? Input c Displays Xa,Ya= (XxY) one solution Prompts POS/NEG? A:P:N (A=all, P=positive, N=negative) Then displays more solutions If a and b has a common divisor you get a&b S.DIV.
DIV	DIVISION	Calculates in infinite number of decimals to a division Prompts N>M Input the numbers to divide Displays quotient R/S starts the calculation. Displays/prints lines of 12 decimals. Input of 2015 and 257: 8 (quotient) 840466926070 038910505836 575875486381 322957198443 579766536964 980544747081 712062256809 338521400778 210116731517 509727626459 ... the first 120 decimals
MUL	MULTIPLY	Multiplies two numbers and gives the accurate result with all digits * uses XF Prompts 00 T Input 1st number press R/S Prompts 01 T Input 2nd number press ON Displays 02 _ result An example: Input 123456, the decimalpoint is crucial ! Uses the current format Input 456789, -ditto- See 563933422784, 123456*456789 = 5,6393342 E10 in normal HP41 mode You are not limited to integers

$L F, T F \quad V + X \{, T \}$

LBL A prompts $L0 \rightarrow V$ Input length₀ and velocity

Displays $LV = \text{length}_v$ length seen at velocity V (shorter)

Input 1m and $2.7 \cdot 10^8 \text{ m/s}$ gives 0.4359 m

LBL B prompts $T1 \rightarrow V$ Input time in rest frame and speed

Displays $T = \text{time}_v$ time seen by stationary observer (longer)

Input 1s and $3 \cdot 10^7 \text{ m/s}$ gives 1.0050 s

LBL C prompts $V1 \rightarrow V2$ Input velocity V_1 and velocity V_2

Displays $V = \text{velocity}$ Sum of velocities

Input $2.9 \cdot 10^8 \text{ m/s}$ and $2.9 \cdot 10^8 \text{ m/s}$ gives 299827685.3 m/s

LBL D prompts $X \rightarrow T \rightarrow V$ Input X and T in stationary system and velocity

Displays $X1 = X$

LBL d prompts $X1 \rightarrow T1 \rightarrow V$ Input X_1 and T_1 in moving system and velocity

Displays $X = X$

LBL E prompts $T \rightarrow X \rightarrow V$ Input T and X in stationary system and velocity

Displays $T1 = \text{time}$

LBL e prompts $T1 \rightarrow X1 \rightarrow V$ Input T_1 and X_1 in moving system and velocity

Displays $T = \text{time}$

LBL J Displays menu

An example:

Lightning simultaneously strikes two trees standing 20 km apart at time 10 μs

At what time will the events be seen from a rocket travelling at $0.5 \cdot c$?

LBL E Input $10 \cdot 10^{-6} \rightarrow 0 \rightarrow 149.90 \cdot 10^6$

and see $T1 = 11.544E-6$ i.e $t_1 = 11.544 \mu\text{s}$

LBL E Input $10 \cdot 10^{-6} \rightarrow 20 \cdot 10^3 \rightarrow 149.90 \cdot 10^6$

and see $T1 = -26.910E-6$ i.e $t_1 = -26.910 \mu\text{s}$

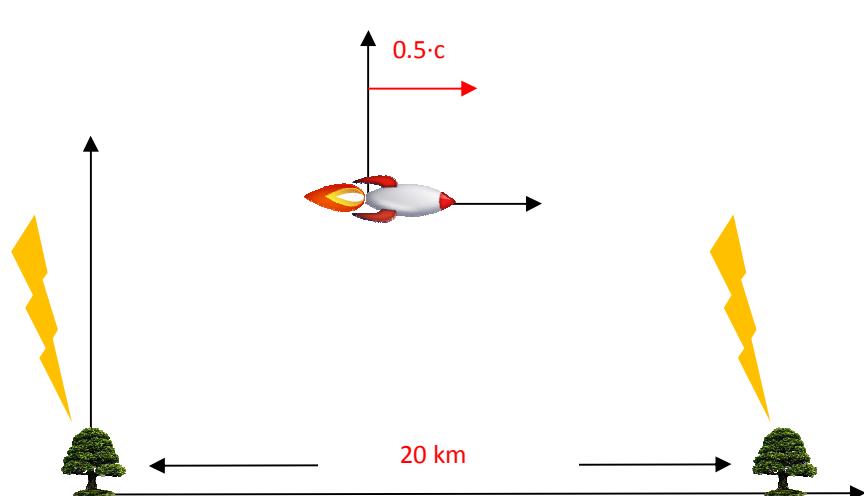
Where will the events be seen from the rocket?

LBL D Input $0 \rightarrow 10 \cdot 10^{-6} \rightarrow 149.90 \cdot 10^6$

and see $X1 = -1.7305E3$ i.e $x_1 = -1.7305 \text{ km}$

LBL D Input $20 \cdot 10^3 \rightarrow 10 \cdot 10^{-6} \rightarrow 149.90 \cdot 10^6$

and see $X1 = 21.358E3$ i.e $x_1 = 21.358 \text{ km}$



an example:

Prompts $X_0?$	Input initial X	1
Prompts $Y_0?$	Input initial Y	2
Prompts $V_0?$	Input initial velocity	10
Prompts $\alpha?$	Input initial angle	45

Displays 

LBL A	Finds y and t for a given x		
	Prompts $X?$	Input x	5
See	$T=t$		$T= 0.5657$
	$Y=y_t$		$Y= 4.4288$

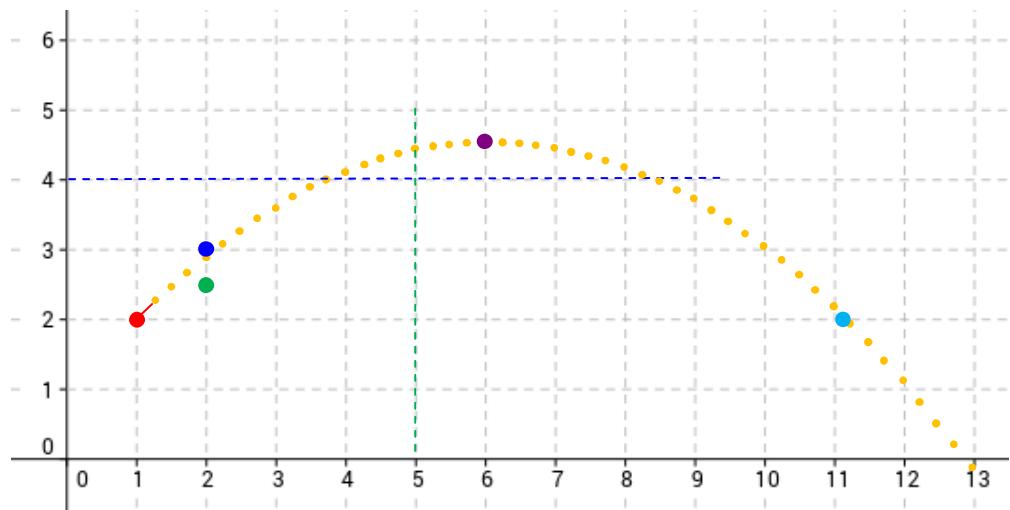
LBL B	Finds x and t for a given y		
	Prompts $Y?$	Input y	4
See	$T=t$		$T= 0.3867$
	$X=x_t$		$X= 3.7340$
and	$T=t$		$T= 1.0535$
	$X=x_t$		$X= 8.4493$

LBL C	Finds x and y for a given t		
	Prompts $T?$	Input t	1
See	$X=x_t$		$X= 8.0711$
	$Y=y_t$		$Y= 4.1611$

LBL D	Finds the angle to hit a target x,y and the time to get there		
	Prompts $X \triangleright Y ?$	Input x,y	2 \triangleright 3
See	$\alpha = \text{angle}_1$		$\alpha = 47.9726$
	$T=t_1$		$T= 0.1494$
and	$\alpha = \text{angle}_2$		$\alpha = 87.0274$
	$T=t_2$		$T= 1.9283$

LBL E	Finds the velocity to hit a target x,y and the time to get there		
	Prompts $X \triangleright Y ?$	Input x,y	2 \triangleright 2.5
See	$V_0 = \text{velocity}$		$V_0 = 4.4317$
	$T=t$		$T= 0.3191$

* LBL D and E stores new values instead of the initial values



	LBL F	Finds the range and the time to get there See R=range R= 10.1833 (light blue dot ●) T=time to get there T= 1.4401
	LBL H	Finds the highest point and the time to get there See X=x X= 6.0916 Y=y _{max} Y= 4.5458 (purple dot ●) T=time T= 0.7201
	LBL J	Displays menu
eENERGI	electron ENERGY	Calculates the energy of electron shells Displays 
	LBL A prompts N?	Input shell #
	Displays E(N)= energy eV	energy state for that shell
	Input of 2 gives E(2)= 10.2000	
	LBL B prompts N1>N2	Input shell #'s
	Displays E(N1,N2)= energy eV	energy released by the transition between shells
	Input of 4 > 2 gives E(4,2)= -2.55 eV	
	LBL C Displays the Lyman series	
	LBL D Displays the Balmer series	
	LBL E Displays the Paschen series	
	LBL J Displays menu	
PENDUL	PENDULUM	Calculates variables for a simple pendulum Displays  Turn on USER!
	LBL A prompts L=?	Input length of the pendulum (m)
	Displays T= time of one oscillation (s)	
	LBL B prompts L=?	Input length of the pendulum (m)
	Displays F= frequency of the oscillation (Hz)	
	LBL C prompts T=?	Input time of one oscillation (s)
	Displays L= length of the pendulum (m)	
AMV	ASTABIL	Calculates frequency and number of blinks/minute for an Astable MultiVibrator
	MULTIVIBRATOR	Displays INPUT MODST. and then prompts Ω KΩ MΩ Input resistor value and press LBL A , LBL C or LBL E
		Displays INPUT KONDENS and then prompts ²F NF PF Input capacitor value and press LBL A , LBL C or LBL E
		Calculates and displays FREK.= frequency Hz
		Press  and see CA. X BL./MIN
		Input of 10 KΩ and 100 ²F gives FREK= 0.7143 Hz and CA. 42 BL./MIN

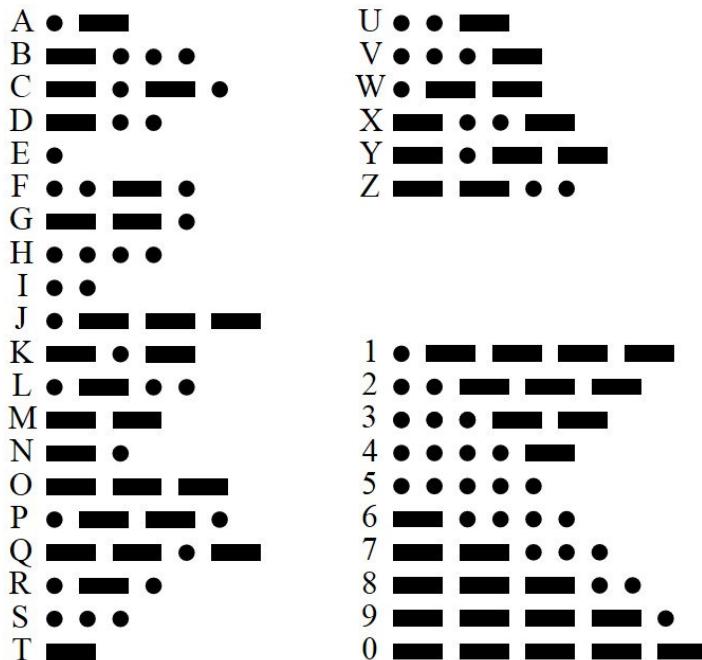
MORSE

MORSE CODE

Plays the Morse Code of a text input

Prompts TEXT: _ input text to code and listen. Then prompts for new text.

Uses an ASCII file named MORSEAL. See next page and below for more



MLOAD

MORSEAL LOAD

Creates the MORSEAL file needed for MORSE

Ángel wrote this program, and it wasn't part of the original collection

INP?

INPUT ?

Helps to input and store a sequence of values in sequential registers

Input a pointer bbb,eeeii and store a name in ALPHA. XEQ INP?

Prompts NAMEbbb? Input the value and press **R/S** This repeats until NAMEeee?

Values are stored in R_{bbb} to R_{eee}

	INPUT TEXT
96	“
153	0
3	←
213	U
3	←
65	A
97	a
113	ä
121	ÿ
125	→
127	†
95	_
79	O
71	G
67	C
143	⌘
3	←
3	←
3	←
3	←
231	¤
3	←
13	≤
47	/
43	+
23	Ø
7	↓
59	:
19	À
63	?
15	⌘
49	1
21	Ä
55	7
9	σ
11	λ
17	Ω
51	ʒ
37	%
27	₪
31	⌘
5	€
29	#
61	=
25	0
45	-
41	>
39	.

PK3

-PK3 TIME		Header ♦ Time functions
DOY	DAY OF YEAR	<p>Calculates what number in year a certain date is * Requires FIX2 and DMY</p> <p>Prompts DATE? Input a date in the form dd,mmyyyy or press R/S for today.</p> <p>The day number is placed in X</p> <p>Input of 11.032015 gives 70</p> <p>i.e. march 11th of 2015 is day number 70 of that year</p>
WOD	WEEK OF DATE	<p>Calculates what week number a certain date is in * Requires DMY format</p> <p>Prompts DATE? Input a date in the form dd,mmyyyy or press R/S for today.</p> <p>The week number is placed in X.</p> <p>Input of 11.032015 gives 11</p> <p>i.e. march 11th of 2015 is in week number 11 of that year</p>
WOW	WEEK OF WEEK	<p>Calculates the dates of a specific week in a specific year</p> <p>Prompts W?D? Input a week number and a year.</p> <p>Displays the start-date and end-date of the week.</p> <p>Input of 9 and 2015 gives 23.02-01.03.15</p>
D?M	DATE OF M-DAY	<p>When is a given date an M-day? * Requires DMY format</p> <p>Prompts DAY? (sun=0, mon=1, tue=2, wed=3, thu=4, fri=5, sat=6)</p> <p>Prompts DATE? Input date in format dd,mmyy</p> <p>Displays the date</p> <p>Input of 2 and 27.022015 gives 27.02.2018 as the next Tuesday on a 27.02</p>
NM?M	Nth M-DAY OF MONTH	<p>When is the Nth M-day of a specific month? * Requires DMY format</p> <p>Prompts DAY? (sun=0, mon=1, tue=2, wed=3, thu=4, fri=5, sat=6)</p> <p>Prompts N? Input the number</p> <p>Prompts M,Y? Input month and year</p> <p>Displays the date</p> <p>Input of 4, 2 and 5,2015 gives 14.15.2015 as the 2nd Thursday in May 2015</p>
M?DD	M-DAYS BETWEEN DATES	<p>How many M-day between two dates? * Requires DMY format</p> <p>Prompts DAY? (sun=0, mon=1, tue=2, wed=3, thu=4, fri=5, sat=6)</p> <p>Prompts DATE1? Input the first date</p> <p>Prompts DATE2? Input the second date</p> <p>Calculates the number</p> <p>Input of 3, 27,022015 and 24,042015 gives 8,00</p> <p>i.e. there are 8 Wednesdays between the dates</p>
CLKSET	CLOCKSET	Sets CLK24 and DMY. Sets the clock and date using X and Y. Runs CLOCK
CLKSEED	CLOCKSEED	Uses the current Time and Date to calculate and store a Seed for the Random function
AUTOCLK	AUTOMATIC CLOCK	Sets F11 and turns the HP41 off. On next wakeup HP41 starts the Clock
SWCLK	STOPWATCH CLOCK	Starts the Stopwatch running with the current time
ADOW	TO ALPHA THE DAY OF WEEK	<p>Puts in Alpha the name of the date in X * Requires DMY format</p> <p>Input 11,032015 XEQ ADOW and see WED in ALPHA</p> <p>i.e. March 11th 2015 is a Wednesday</p>

DDATE	DAYS FROM DATE	Calculates and displays X days from the date in Y * Requires DMY format Input of 28,022015 and 50 displays 19.04.15 SUN																					
		<table border="1"> <thead> <tr> <th></th><th>X</th><th>Y</th><th>Z</th><th>T</th><th>L</th><th>ALPHA</th></tr> </thead> <tbody> <tr> <td>Before</td><td>50</td><td>28,022015</td><td></td><td></td><td></td><td>ALPHA</td></tr> <tr> <td>After</td><td>19,042015</td><td>0</td><td>SUN</td><td>19,042015</td><td>0</td><td>SUN</td></tr> </tbody> </table>		X	Y	Z	T	L	ALPHA	Before	50	28,022015				ALPHA	After	19,042015	0	SUN	19,042015	0	SUN
	X	Y	Z	T	L	ALPHA																	
Before	50	28,022015				ALPHA																	
After	19,042015	0	SUN	19,042015	0	SUN																	
DOE	DATE OF EASTER	Calculates and displays the date of Easter of the year in X Input 2015 XEQ DOE displays 5.apr 2015																					
FCNTIME	FUNCTION TIME	Finds the execution time of any function or program Create a program named T and the program line(s) should be the function(s) you want to test. XEQ FCNTIME																					
		Displays  or similar The function time is 0.24 second																					
DOD	DATE OF DAY	Calculates the date of a day number in a year Prompts Y>D input year and day number Input of 2015 and 70 gives 11.032015 i.e. march 11th 2015																					
C-J	CALENDAR TO JULIAN DAY NUMBER	Converts a calendar date to Julian Day number Input 1985,0217 and see 2 444 113.5																					
J-C	JULIAN DAY NUMBER TO CALENDAR	Converts a Julian Day number to a calendar date Input 1842713 and see 333.01275 i.e. january 27th year 333																					

TMP TEMPERATURE Converts between 5 temperature scales * USER must be turned ON

Put the value to convert into X. XEQ TMP

Displays



Press the scale you are converting from. The menu re-appears

Press the scale you are converting to. The new value is in X

For continued calculations press LBL J. For a new calculation input X and press LBL J

LBL A	LBL B	LBL C	LBL D	LBL E
Celcius	Fahrenheit	Réaumur	Rankine	Kelvin

An example:

Convert 80°C to Fahrenheit:

Input 80 XEQ TMP press A press B see 176.00

Convert 176°F to Kelvin:

R>D ROMAN TO DIGITAL Converts a Roman number to a digital number

Prompts R'= _ for input of the Roman number. Displays D'= digital.

Digital number is in X

Input of MMXV displays D'= 2015

D>R DIGITAL TO ROMAN Converts a digital number to a Roman number * uses CCD and XF module

Prompts D'= for input of the digital number. Displays R'= Roman.

Roman number is stored in an ASCII file called R

Input of 2015 displays R'= MMXV

10-BASE BASE10 TO BASE IN Converts a digital number in X to the base in Y

Y Displays X in BASEy X and Y are preserved

Input of 4 ↗ 255 displays 3333

BIN-BCD BINARY TO BCD Converts the Hex code of the 4 rightmost digit in X to decimal number

BCD-BIN BCD TO BINARY Converts the decimal number in X to Binary

CLDK	CLEAR DIGIT KEYS	Clears the assignments on keys - 7 8 9, + 4 5 6, × 1 2 3 and ÷ 0 . R/S (only top keys)
CLMK	CLEAR MENU KEYS	Clears the assignments on keys A-E, a-e and F-J
CLBF	CLEAR BUFFER	Clears the Buffer area (Buffers & Keys assignments)
CLXM	CLEAR EXTENDED MEMORY	Clears XM
CLMM	CLEAR MAIN MEMORY	Clears main memory, key assignments etc.
CLPRGMS	CLEAR PROGRAMS	Clears programs in RAM
SUSK	SUSPEND KEYS	Stores the Keys Assignment in a buffer
ACTK	ACTIVATE KEYS	Restores the Keys Assignment from buffer
PACKK	PACK KEYS	Reclaims unused space in the Keys Assignment area
CPR	CPR NUMBER	<p>Tests if the input number is a valid danish CPR number (Personal Identification Number) http://en.wikipedia.org/wiki/Personal_identification_number_(Denmark)</p> <p>Prompts _____ - _____</p> <p>The first six digits are your birthday in the form ddmmyy You can't input a nonexisting date!</p> <p>The last four digits is a sequence number (last digit is odd for males and even for females) Program tests for validity and displays LEGAL CPR# or ILLEGAL CPR# 160315-0127 would qualify as a legal CPR# for a male born on 16th of March 2015</p>

PK4

-PK4 PRGM Header • Programming functions		
A?	ASSIGNMENT SIZE?	Returns the number of registers used by key assignments
B?	BUFFER SIZE?	Returns the size of the buffer in X
C?	CURTAIN?	Returns the absolute address of the 'Curtain'
D?	DECIMALS?	Returns the number of decimals displayed
E?	END?	Returns the absolute address of the .END.
F?	FREE REGISTERS?	Returns the number of free registers in memory
I?	ROM ID?	Displays ROM ID of the ROM# in X
L?	LINE?	Returns the line number in actual program
P?	PROGRAM SIZE?	Returns the RAM size of the current program
R?	REVISION?	Returns the revisions of the HP41 ROM0,ROM1,ROM2. Displays ROM=NFL or similar
S?	SIZE?	Returns the number of registers allocated for storage
W?	WORD SIZE?	Returns the Word size used by binary functions
Σ?	Σ?	Returns the absolute address of ΣREG00
CAT_---	CAT_---	Runs Catalog 3 starting at the function# indicated by your input
AFCN?	ALPHA FUNCTION?	Displays information about a function: Prompts AFCN? _ for a function name  Brings up the next line of information.  aborts the function
If input is a native HP41 function (like DEC) you get:		
	ADR= 132B	the functions address i HP41ROM
	DEC= 004,095	the decimal value input to get DEC
	HEX= 04,5F	the hexadecimal value to get DEC
	MAINFRAME	indicates a native HP41 function
If input is in a module (like AFCN?) you get:		
	ADR= B295	the functions address i ROM
	DEC= 161,078	the decimal value input to get AFCN?
	HEX= A1,4E	the hexadecimal value to get AFCN?
	XROM 05,14	XROM# for AFCN?
?BUF	? BUFFER EXISTS	Prints a list of the existing buffer IDs * uses HEPAX (also see Technical Stuff page 60)
(c)	(COPYRIGHT)	Displays (c) 1990 POUL KAARUP
TPRV	TOGGLE PRIVATE	Makes or un-makes a program PRIVATE Prompts TRPV _ Input program name or press ALPHA twice for the current program The program will be made PRIVATE, or if it already is, it will be made UNPRIVATE
FATCAT	FAT CATALOG	Displays/prints addresses and function names of FAT * uses HEPAX Prompts FAT ADR: _ _ _ Input C002 and see what's in Page C:
		C08F -PK1a ALPHA C111 ADL C0F0 ADR C4B9 AIX C0E6 AXL C0EB AXR C4D9 AXX C120 XAL

JUMP1

JUMP DISTANCE

FOR A CLASS 1

JUMP

Calculates the mcode for a Class 1 jump

Prompts ?_

For ?C GO ____ press  and then 

For ?C XQ ____ press  and then 

For ?NC GO ____ press  and then 

For ?NC XQ ____ press  and then 

 Lets you go back one step

Input the address ____ HEX!

and get the words needed for the jump

Input ?NC XQ A4B5 displays HEX = 2D5-293 i.e. the words to program are 2D5 and 293

JUMP3

JUMP DISTANCE

FOR A CLASS 3

JUMP

Calculates the mcode for a Class 3 jump

Prompts J_

For JC- ____ press  and then 

For JC+ ____ press  and then 

For JNC- ____ press  and then 

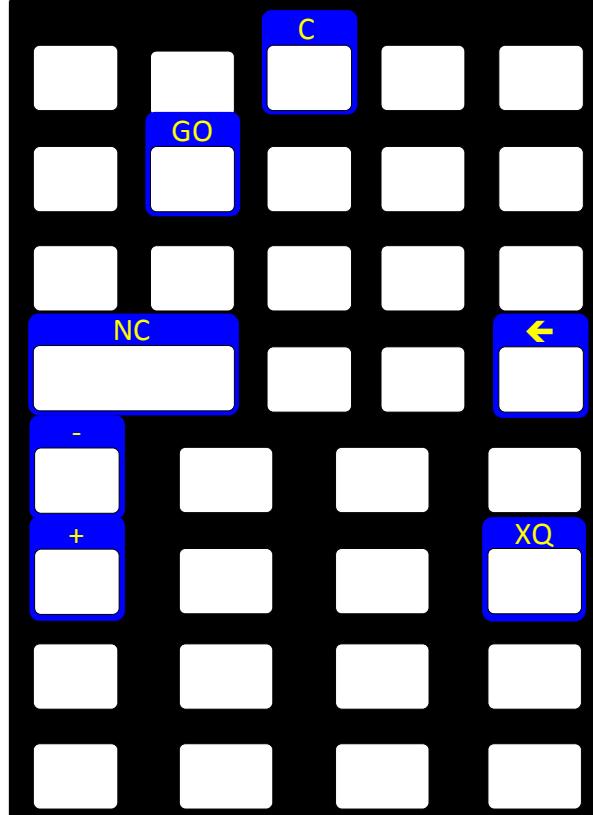
For JNC+ ____ press  and then 

 Lets you go back one step

Input the jump distance ____ HEX!

and get the word needed for the jump

Input JNC+ 2F displays HEX= 17F i.e. the word to program is 17F



FL?	FILE NAME?	Prompts for input of a filename Prompts FIL: _____ Input name (is stored in ALPHA)
SAVERTN	SAVE RETURN STACK	Saves the return stack (Reg a + Reg b) to a buffer The return stack holds 6 RTNs. If you need more, you can use SAVERTN to save the return stack and have a clean stack with 6 new returns. Use GETRTN to retrieve the return stack
GETRTN	GET RETURN STACK	Copies the return stack in buffer to Reg a and Reg b
CURTAIN	SET CURTAIN	Sets the 'Curtain' address Prompts CURTAIN ____ Input address
PPACK	PROGRAMMABLE PACK	Packs program memory
GE	GO TO .END.	Places program pointer at .END. Like GTO... but no packing
CREND	CREATE END	Creates END without packing
CSST	CONTINUOUS SST	Singlesteps automatically through current program * If FS00 it goes backwards



Aborts the viewing



Reverses direction



All other keys affect the speed

S+ is fastest

USER is slowest

→ ALPHA → PRGM → USER

HXENTRY	HEXADECIMAL ENTRY	Allows you to enter a Hexadecimal number. Prompts _____ Decimal point and R/S puts the number in X as NNN
GORAM	GO TO RAM	Moves the program pointer to FOCAL program in RAM Address must be in X, and can be obtained with GETPC from ASSEMBLER3 or RCL b

DEBUG	DEBUG PROGRAM	Inserts a break-point in an mcode program and halts execution at that point, allowing you to see the CPU registers and pointers * See Ken Emery's book for more information
LOOP	LOOP DEBUG PROGRAM	Allows you to debug a loop within an mcode program * See Ken Emery's book for more information
RSLCT	RAMSLCT	Allows you to see the RAMSLCT pointer and the T register * See Ken Emery's book for more information

ABS reg #	nybbles																			
	13	12	11	10	09	08	07	06	05	04	03	02	01	00						
487	0	0	RTN 3				RTN 2				RTN 1									
488	KY	RTN 4				XY	P	Q					G	ST						
CPU register C																				
490	CPU register A																			
491	CPU register B																			
492	CPU register M																			
493	CPU register N																			
494	STATUS register T																			
495	STATUS register Z																			
496	STATUS register Y																			
497	STATUS register X																			
498	STATUS register L																			
499	STATUS register M																			
500	STATUS register N																			
501	STATUS register O																			
502	STATUS register P																			
503	STATUS register Q																			
504	STATUS register $\text{P}^{\text{-}}$																			
505	STATUS register a																			
506	STATUS register b																			
507	STATUS register c																			
508	STATUS register d																			
509	STATUS register e																			
510	BREAK address							word												
511	BREAK address+1							word												
	#	13	12	11	10	09	80	07	06	05	04	03	02	01	00					

XY bit	07	06	05	04	03	02	01	00
	CPU flag #	13	12	11	10	9	8	v

↓ ↓

0 =	hex mode	SLCT P
1 =	dec mode	SLCT Q

PEEK	PEEK	Allows you to see register in RAM Prompts <u>___</u> Puts contents of that (absolute) address in X
POKE	POKE	Allows you to change in RAM Prompts <u>___</u> Puts X to that (absolute) address
OCT-HEX	OCTAL TO HEXADECIMAL	Converts mainframe addresses to Hexadecimal address Prompts O _ Input page number (0-7) Prompts O 7- _ Input quad number (0-3) Prompts O 7-3- _ Input address (0000-1777) Displays ADDRESS 7FFF
HEX-OCT	HEXADECIMAL TO OCTAL	Converts Hexadecimal addresses to octal Prompts H _ Input address (0000-FFFF) Displays OCT F-3-1777
XQ>XR	XEQ TO XROM	Will change all XEQ's to XROM's where possible (modules are present) Keyed in while the module wasn't present:  Will look like this afterwards: 
UPDFAT	UPDATE FAT	Put name of program to be revised in ALPHA. If ALPHA clear it uses current program Updates FAT when you have keyed in a new routine Put address of first executable word in ALPHA, like 81F3 XEQ CODE XEQ UPDFAT Increments the number of functions and adds the routine address to end of FAT
UPFAT	UPDATE FAT	Appends address to FAT when you have keyed in a new routine Address of first executable word must be in R00 then XEQ UPDFAT Is mainly to be used after using ASSEM in ASSEMBLER3
APPLBL	APPEND LABELS	Reads location of every ALPHA-label in FOCAL program and appends the addresses to FAT Loading address in Hex digits right justified must be in X Is mainly to be used with LOADP in ASSEMBLER3

XTO3N	X TO 3 Nybbles	Converts a number in X to three hexadecimal nybbles appended to ALPHA rightmost Input X (0-4095) see nybbles in ALPHA
3NTOX	3 Nybbles TO X	Converts the three rightmost nybbles in ALPHA to a number in X Range X (0-4095) and nybbles removed from ALPHA
XTON	X TO Nybble	Converts a number in X to one hexadecimal nybble appended to ALPHA rightmost Input X (0-15) see nybble in ALPHA
NTOX	Nybble TO X	Converts the rightmost nybbles in ALPHA to a number in X Range X (0-15) and nybble removed from ALPHA
INCRM	INCREMENT M	Increments the pointer at the three rightmost nybbles in M
DECRM	DECREMENT M	Decrements the pointer at the three rightmost nybbles in M
DATED	DATA FILE EDIT	Tool to edit a DATA file in extended memory Displays  LBL A Displays contents of next register R5= R05 _{XM} LBL B Displays contents of previous register R3= R03 _{XM} LBL C Prompts R=? Input number of register to edit LBL D Prompts FIL: _ Input name of data file to edit LBL E Displays contents of current register R4= R04 _{XM} LBL J Displays menu
VWFL	VIEW FILE	Views the contents of an ASCII-file in XM * uses CCD module If FS 02: Prompts FIL: _ Input name of ASCII-file If FC 02: Uses active file
PRFL	PRINT FILE	Prints the contents of an ASCII-file If FS 02: Prompts FIL: _ Input name of ASCII-file If FC 02: Uses active file
WRTASFL	WRITE ASCII FILE	Creates a file on HP-IL Drive and copies an ASCII-file from XM to HP-IL If FS 02: Prompts FIL: _ Input name of ASCII-file If FC 02: Uses active file
WF?	WORKFILE TYPE ?	Returns a number to identify the type of the current file in XM If No Workfile: 0 If ASCII file: 1 If Data file: 2

RAMED

RAM EDIT

Tool to edit RAM memory byte by byte * uses CCD module

**LBL A** Displays next byte**LBL a** Displays next register**LBL B** Displays previous byte**LBL b** Displays previous register**LBL C** Prompts BYTE (bb) ' __ Input new byte**LBL D** Prompts FIL: Input name of program file to edit**LBL E** Prompts ADR,BYTE? Input absolute address in RAM**LBL J** Displays menu*An example:*At first we make a program named TEST with the lines SIN and STO 01
XEQ RAMED

LBL D	FIL:	Input TEST
	411.6 00:C0:00	byte 6 of reg 411 is C0
LBL b	previous reg	410.6 59:54:53
LBL B	previous byte	410.5 31:59:54
LBL C	poke new byte	BYTE (59) ' __
		Input 5A (COS)
		410.5 31:5A:54
LBL A	press twice	411.0 54:53:45
LBL C	poke new byte	BYTE (53) ' __
LBL B	press 3 times	410.4 C6:31:5A
LBL C	poke new byte	BYTE (31) ' __
		Input 32 (STO 02)

Go to program TEAT and see the lines COS and STO 02

PRRAM

PRINT RAM

Prints a program in RAM memory * uses CCD module

If FS 02: Prompts for filename FIL: __ Input TEAT**If FC 02:** Uses name in ALPHA

Printout:

```

        411.6000    ***
       6 5 4 3 2 1 0
1 C000F500544541 @+u*TEA
0 545A32C6010900 TZ2F*x*
9 00000000C20120 ***+B*
       6 5 4 3 2 1 0
        408.6000    ***

```

F&R

FIND & REPLACE

Finds and replaces characters in an ASCII-file in XM * uses CCD module

If FS 02: Prompts FIND: __ Input **text₁**Prompts REPLACE: __ Input **text₂**Prompts FIL: __ Input **name of ASCII-file****If FC 02:** it can be used as a subroutine (ASCII-file must be active file):**length of text₁** STO L**text₁** ASTO X**text₂** ASTO Y

FC-XR	FCODE TO XROM	Converts a hexadecimal code from a FOCAL program in ROM to an XROM number Prompts 1__ 0__ Input 1A7 0C2 and see XROM 31.02
XR-FC	XROM TO FCODE	Converts an XROM number to hexadecimal code in a FOCAL program in ROM Prompts ___.__. Input XROM 31.02 and see 1A7 0C2

* These two programs were needed during the process of restoring my modules, and will be in the next edition of this collection (I hope)

Ax xx

1010 0ddd ddff ffff

1010 = A

ddddd = device#
fffff = function#

Some Formulas

KOMB LBL A
$$P(n, p) = \frac{n!}{(n - p)!}$$

LBL B n^p

LBL C
$$C(n, p) = \frac{n!}{p!(n - p)!}$$

LBL D $C(n+p-1, p)$

LBL E
$$\frac{C(n - x, r) \cdot r}{C(n, r) \cdot (n - x)}$$

LBL F
$$\frac{N!}{(s_1!)^{n_1!} \cdot n_1! \cdot (s_2!)^{n_2!} \cdot n_2! \cdot \dots \cdot (s_i!)^{n_i!} \cdot n_i!}$$

LBL G
$$\frac{n!}{r_1! \cdot r_2! \cdot r_3! \cdot \dots}$$

LBL H
$$\frac{\frac{R!}{x!(R-x)!} \cdot \frac{(N-R)!}{x!(N-R-x)!}}{N!} \\ \frac{}{U!(N-n)!}$$

LBL I
$$\frac{k!}{x!(k-x)!} \cdot p^x \cdot (1-p)^{k-x}$$

LBL d $C(m_1, x_1) \cdot C(m_2, x_2) \cdot \dots \cdot C(m_i, x_i)$

LBL e
$$\frac{n! \cdot p_1^{x_1} \cdot p_2^{x_2} \cdot \dots \cdot p_k^{x_i}}{x_1! \cdot x_2! \cdot \dots \cdot x_i!}$$

LORENTZ

LBL A

$$L_v = L_0 \cdot \sqrt{1 - \frac{v^2}{c^2}}$$

LBL B

$$t_v = \frac{t}{\sqrt{1 - \frac{v^2}{c^2}}}$$

LBL C

$$v = \frac{v_1 + v_2}{1 + \frac{v_1 \cdot v_2}{c^2}}$$

LBL D

$$x_1 = \frac{x - v \cdot t}{\sqrt{1 - \frac{v^2}{c^2}}}$$

LBL E

$$t_1 = \frac{t - \frac{vx}{c^2}}{\sqrt{1 - \frac{v^2}{c^2}}}$$

LBL d

$$x = \frac{x_1 + v \cdot t_1}{\sqrt{1 - \frac{v^2}{c^2}}}$$

LBL e

$$t = \frac{t_1 + \frac{vx}{c^2}}{\sqrt{1 - \frac{v^2}{c^2}}}$$

HERON $A = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}$

TMP	Fahrenheit	${}^{\circ}\text{F} = {}^{\circ}\text{C} \cdot 1.8 + 32$
	Réamur	${}^{\circ}\text{Re} = {}^{\circ}\text{C} \cdot 0.8$
	Rankine	${}^{\circ}\text{Ra} = ({}^{\circ}\text{C} + 273.15) \cdot 1.8$
	Kelvin	$\text{K} = {}^{\circ}\text{C} + 273.15$

PYTH

$$\begin{aligned} a^2 + b^2 &= c^2 \\ a &= p \cdot (m^2 - n^2) \\ b &= p \cdot 2 \cdot m \cdot n \\ c &= p \cdot (m^2 + n^2) \end{aligned}$$

PB LBL A $t = \frac{x - x_0}{v_0 \cdot \cos \theta}$

$$y_t = y_0 + v \cdot t \cdot \sin(\theta) - \frac{1}{2} \cdot g \cdot t^2$$

LBL B $t = \frac{v \cdot \sin \theta \pm \sqrt{v^2 \sin^2 \theta + 2 \cdot g(y_0 - y_t)}}{g}$

$$x_t = x_0 + v \cdot t \cdot \cos(\theta)$$

LBL C $x_t = x_0 + v \cdot t \cdot \cos(\theta)$
 $y_t = y_0 + v \cdot t \cdot \sin(\theta) - \frac{1}{2} \cdot g \cdot t^2$

LBL D $\theta = \tan^{-1} \left(\frac{v^2 \pm \sqrt{v^4 - g \cdot (gx^2 + 2yv^2)}}{gx} \right)$

LBL E $v = \sqrt{\frac{g \cdot (y - y_0)^2}{2 \cdot \cos^2 \theta \cdot (\tan \theta \cdot (y - y_0) - (x - x_0))}}$

$$t = \frac{x - x_0}{v_0 \cdot \cos \theta}$$

LBL F $d = \frac{v^2}{g} \cdot \sin 2\theta$

LBL H $t_h = \frac{v_0 \cdot \sin \theta}{g}$

$$h = v_0 \cdot t_h \cdot \sin \theta - \frac{1}{2} \cdot g \cdot t_h^2$$

LBL A $\sin^{-1} z = -i \cdot \ln(i z + \sqrt{1 - z^2})$

LBL B $\cos^{-1} z = -i \cdot \ln(z + i \sqrt{1 - z^2})$

LBL C $\tan^{-1} z = \frac{i}{2} \ln\left(\frac{i+z}{i-z}\right)$

LBL a $\sinh^{-1} z = \ln(z + \sqrt{z^2 + 1})$

LBL b $\cosh^{-1} z = \ln(z + \sqrt{z^2 - 1})$

LBL c $\tanh^{-1} z = \frac{1}{2} \ln\left(\frac{1+z}{1-z}\right)$

Technical Stuff

XROM ID's

PK1a	PK1b	PK2c	PK2d	PK3e	PK3f
31	3	14	15	18	5

Label-less

In a number of programs you can find a GTO or XEQ that doesn't have a corresponding LBL. The jump-distance is 'hidden' in the GTO or XEQ and therefore no LBL is needed. Advantage: saves space. Dis-advantage: makes it difficult to decipher what the program actually does (trust me!).

I think the label-less is also responsible for the fact that it makes rubbish when I COPY programs to RAM (on i41CX+ and V41) and then try to edit them.

Names

Some of the names of programs and functions can't be keyed in using XEQ. This may seem silly, but it was part of the "*if it can be done, then do it*"-project exploring mcode.

You can assign and execute them using the CCD module catalog function, or you can use the second set of modules, where we've changed the function names, so they don't look right to me (& Ángel) but they can be keyed in normally.

Doubles?

Why are some of the functions in two ROMs ?

Originally my programs were in three ZEPROM modules. Because there are only 4 slots in the HP41 and that the printer, the Tape-drive, HEPAX and the CCD are heavily used in the programs, there often wasn't room for the extra module. We have also included some functions from ASSEMBLER3 and CCD to minimize this problem.

Buffer ID's

ID	Module	Buffer stores:
1	David Assembler	Mcode labels already existing
2	David Assembler	Mcode labels referred to
3	Eramco RSU-1B	ASCII data pointers
4	Eramco RSU-1A	Data file pointers
5	CCD / Advantage	Word size + Seed / Matrix name
6	Extended IL	Accessory ID of current device
7	PK3e & PK3f / Extended IL	RTN Stack & Key Assignments / Printing column number & width
8	41Z	Complex Stack and Mode
9	SandMath / PowerCL	Seed / Last function data
10	TIME	Alarms
11	Plotter	Data & Barcode parameters
12	CMT-200 / HPIL-DEV	IL Buffer / I/O monitoring
13	CMT-300 / FORTH	Status info / FORTH buffer
14	Advantage / SandMath	INTEG & SOLVE scratch
15	HP41	Key assignments

i41CX+

This amazing emulator for iOS was used to create the screendumps and printouts and for most of the testing.

XROM numbers	XROM 31.00	-PK1a ALPHA	XROM 03.00	-PK1b FLAGS	XROM 03.57	-PK1b TABLE
mcode	XROM 31.01	ADL	XROM 03.01	FLAGMAP	XROM 03.58	CONSTAN
FOCAL	XROM 31.02	ADR	XROM 03.02	F0	XROM 03.59	GEOPOS
	XROM 31.03	ADX	XROM 03.03	F1	XROM 03.60	PERIOD
	XROM 31.04	AXL	XROM 03.04	F2	XROM 03.61	EXP
	XROM 31.05	AXR	XROM 03.05	F3	XROM 03.62	MLOAD
	XROM 31.06	AXX	XROM 03.06	F4	XROM 03.63	VNC
	XROM 31.07	XAL	XROM 03.07	F5		
	XROM 31.08	XAR	XROM 03.08	F6		
	XROM 31.09	XAY	XROM 03.09	F7		
	XROM 31.10	XSY	XROM 03.10	F8		
	XROM 31.11	XAI	XROM 03.11	F9		
	XROM 31.12	YAI	XROM 03.12	F>X		
	XROM 31.13	A=X	XROM 03.13	X>F		
	XROM 31.14	A=Y	XROM 03.14	FC?S		
	XROM 31.15	VWA	XROM 03.15	FS?S		
	XROM 31.16	WAP	XROM 03.16	CFX		
	XROM 31.17	VWT	XROM 03.17	SFX		
	XROM 31.18	VW%	XROM 03.18	TFX		
	XROM 31.19	A(X)=Y	XROM 03.19	ENG		
	XROM 31.20	A(XY)=Z	XROM 03.20	FIX		
	XROM 31.21	A(IX)=Y	XROM 03.21	SCI		
	XROM 31.22	A=(XY)	XROM 03.22	F/E		
	XROM 31.23	X Y	XROM 03.23	FIX/ENG		
	XROM 31.24	STAK	XROM 03.24	DIG		
	XROM 31.25	TLC	XROM 03.25	-PK1b STK		
	XROM 31.26	REG>A	XROM 03.26	X>A>		
	XROM 31.27	A>REG	XROM 03.27	Y<>Z		
	XROM 31.28	A A	XROM 03.28	Y<>T		
	XROM 31.29	ANL	XROM 03.29	Z<>T		
	XROM 31.30	ANR	XROM 03.30	STC		
	XROM 31.31	ALF	XROM 03.31	STR		
	XROM 31.32	A<>	XROM 03.32	STS		
	XROM 31.33	CON	XROM 03.33	STT		
	XROM 31.34	JN	XROM 03.34	STV		
	XROM 31.35	GO	XROM 03.35	STX		
	XROM 31.36	1L	XROM 03.36	X>=Y?		
	XROM 31.37	2L	XROM 03.37	X>=0?		
	XROM 31.38	3L	XROM 03.38	X=Z?		
	XROM 31.39	DISPLAY	XROM 03.39	X=T?		
	XROM 31.40	-PK1a POINT	XROM 03.40	X=1?		
	XROM 31.41	BPT	XROM 03.41	Y=Z?		
	XROM 31.42	BPTI	XROM 03.42	Y=T?		
	XROM 31.43	BPTR	XROM 03.43	Z=T?		
	XROM 31.44	CLI	XROM 03.44	X>Z?		
	XROM 31.45	PTR	XROM 03.45	X<Z?		
	XROM 31.46	-PK1a REG'S	XROM 03.46	X>T?		
	XROM 31.47	RCL	XROM 03.47	X<T?		
	XROM 31.48	STO	XROM 03.48	X>1?		
	XROM 31.49	X<>	XROM 03.49	X<1?		
	XROM 31.50	VIEW	XROM 03.50	X=YZ?		
	XROM 31.51	ST>L	XROM 03.51	X=YZT?		
	XROM 31.52	ST>O	XROM 03.52	-PK1b TONE		
	XROM 31.53	RC>L	XROM 03.53	BEEPMOD		
	XROM 31.54	RC>O	XROM 03.54	BUZZMOD		
	XROM 31.55	P<>S	XROM 03.55	TONEX		
	XROM 31.56	DSZ	XROM 03.56	TONEXY		
	XROM 31.57	ISZ				
	XROM 31.58	PAPER				
	XROM 31.59	?				
	XROM 31.60	(c)				

XROM numbers	XROM 14.00	-PK2c MATH	XROM 14.60	aX+bY=c	XROM 15.00	-PK2d PHYS
mcode	XROM 14.01	SINH	XROM 14.61	X%Y	XROM 15.01	LORENTZ
FOCAL	XROM 14.02	COSH	XROM 14.62	MUL	XROM 15.02	eENERGI
	XROM 14.03	TANH	XROM 14.63	DIV	XROM 15.03	PB
	XROM 14.04	ASINH			XROM 15.04	PENDUL
	XROM 14.05	ACOSH			XROM 15.05	AMV
	XROM 14.06	ATANH			XROM 15.06	MORSE
	XROM 14.07	CBRT			XROM 15.07	INP?
	XROM 14.08	X ³			XROM 15.08	FLAG1
	XROM 14.09	BFACT			XROM 15.09	FLAG2
	XROM 14.10	BY ³ X			XROM 15.10	FLAG3
	XROM 14.11	QR			XROM 15.11	FLAG4
	XROM 14.12	GCD				
	XROM 14.13	LCM				
	XROM 14.14	DIVISOR				
	XROM 14.15	PRIME?				
	XROM 14.16	NXTPRIM				
	XROM 14.17	INTY				
	XROM 14.18	FRCY				
	XROM 14.19	UNARY				
	XROM 14.20	VWM				
	XROM 14.21	Σ				
	XROM 14.22	Σ1				
	XROM 14.23	Σ2				
	XROM 14.24	Σ3				
	XROM 14.25	KOMB				
	XROM 14.26	K				
	XROM 14.27	P				
	XROM 14.28	KEGLESN				
	XROM 14.29	R				
	XROM 14.30	R1				
	XROM 14.31	R2				
	XROM 14.32	V				
	XROM 14.33	LGI				
	XROM 14.34	PC				
	XROM 14.35	PE				
	XROM 14.36	CHT				
	XROM 14.37	CP2				
	XROM 14.38	()*()				
	XROM 14.39	PTEGN				
	XROM 14.40	XY				
	XROM 14.41	LR				
	XROM 14.42	R-S				
	XROM 14.43	S-R				
	XROM 14.44	dB+				
	XROM 14.45	dB-				
	XROM 14.46	PG1				
	XROM 14.47	PG2				
	XROM 14.48	PGA				
	XROM 14.49	POLY				
	XROM 14.50	2POLY				
	XROM 14.51	3K				
	XROM 14.52	PYTH				
	XROM 14.53	HERON				
	XROM 14.54	CIRKEL				
	XROM 14.55	CIRKTAN				
	XROM 14.56	AFSTAND				
	XROM 14.57	LINIE				
	XROM 14.58	3AREAL				
	XROM 14.59	Δ				

XROM numbers	XROM 18.00	-PK3e TIME	XROM 18.57	-CCD FCN'S	XROM 05.00	-PK3f PRGM
mcode	XROM 18.01	DOY	XROM 18.58	ABSP	XROM 05.01	A?
FOCAL	XROM 18.02	WOD	XROM 18.59	ARCLI	XROM 05.02	B?
	XROM 18.03	WOW	XROM 18.60	CLA-	XROM 05.03	C?
	XROM 18.04	D?M	XROM 18.61	KEY?	XROM 05.04	D?
	XROM 18.05	NM?M	XROM 18.62	PMTA	XROM 05.05	E?
	XROM 18.06	M?DD	XROM 18.63	PMTK	XROM 05.06	F?
	XROM 18.07	CLKSET			XROM 05.07	I?
	XROM 18.08	CLKSEED			XROM 05.08	L?
	XROM 18.09	AUTOCLK			XROM 05.09	P?
	XROM 18.10	SWCLK			XROM 05.10	R?
	XROM 18.11	ADOW			XROM 05.11	S?
	XROM 18.12	DDATE			XROM 05.12	W?
	XROM 18.13	DOD			XROM 05.13	N?
	XROM 18.14	DOE			XROM 05.14	AFCN?
	XROM 18.15	FCNTIME			XROM 05.15	FATCAT
	XROM 18.16	-PK3e CONV			XROM 05.16	CAT3 _--
	XROM 18.17	10-BASE			XROM 05.17	JUMP1
	XROM 18.18	BCD-BIN			XROM 05.18	JUMP3
	XROM 18.19	BIN-BCD			XROM 05.19	DEBUG
	XROM 18.20	D>R			XROM 05.20	LOOP
	XROM 18.21	R>D			XROM 05.21	RSLCT
	XROM 18.22	TMP			XROM 05.22	PEEK
	XROM 18.23	J-C			XROM 05.23	POKE
	XROM 18.24	C-J			XROM 05.24	OCT-HEX
	XROM 18.25	F11			XROM 05.25	HEX-OCT
	XROM 18.26	-PK3e REG			XROM 05.26	10-BASE
	XROM 18.27	STO			XROM 05.27	HXENTRY
	XROM 18.28	ST+			XROM 05.28	INCRM
	XROM 18.29	ST-			XROM 05.29	DECRM
	XROM 18.30	ST/			XROM 05.30	XTO3N
	XROM 18.31	ST*			XROM 05.31	3NTOX
	XROM 18.32	ST>L			XROM 05.32	XTON
	XROM 18.33	ST>0			XROM 05.33	NTOX
	XROM 18.34	RCL			XROM 05.34	XQ>XR
	XROM 18.35	RC+			XROM 05.35	APPLBL
	XROM 18.36	RC-			XROM 05.36	UPFAT
	XROM 18.37	RC/			XROM 05.37	UPDFAT
	XROM 18.38	RC*			XROM 05.38	CSST
	XROM 18.39	RC>L			XROM 05.39	CREND
	XROM 18.40	RC>0			XROM 05.40	GE
	XROM 18.41	X<>			XROM 05.41	TPRV
	XROM 18.42	VIEW			XROM 05.42	PPACK
	XROM 18.43	P<>S			XROM 05.43	GORAM
	XROM 18.44	-PK3e UTIL			XROM 05.44	?BUF
	XROM 18.45	CPR			XROM 05.45	CURTAIN
	XROM 18.46	AP			XROM 05.46	SAVERTN
	XROM 18.47	ACTK			XROM 05.47	GETRTN
	XROM 18.48	SUSK			XROM 05.48	DATED
	XROM 18.49	PACKK			XROM 05.49	RAMED
	XROM 18.50	CLBF			XROM 05.50	PRRAM
	XROM 18.51	CLDK			XROM 05.51	PRFL
	XROM 18.52	CLMK			XROM 05.52	VWFL
	XROM 18.53	CLMM			XROM 05.53	WRTASFL
	XROM 18.54	CLPRGMS			XROM 05.54	F&R
	XROM 18.55	CLXM			XROM 05.55	FL?
	XROM 18.56	VWA			XROM 05.56	WF?
					XROM 05.57	DIGST
					XROM 05.58	APPFN
					XROM 05.59	2-D

