free42 Base-N Tools

Mitch Richling

2021-03-19

Author: Mitch Richling
Updated: 2021-05-18 21:09:45

Contents

1 Metadata

The home for this HTML file is: `https://richmit.github.io/hp42/base.html`
A PDF version of this file may be found here: `https://richmit.github.io/hp42/base.pdf`
Files related to this document may be found on github: `https://github.com/richmit/hp42`
Directory contents:

`src`	-	The org-mode file that generated this HTML document
`src_42s`	-	Ready to convert source listings for 42s code in this document
`docs`	-	This html document and associated PDF
`bin`	-	Importable RAW program files

2 Introduction

When I'm doing embedded programming I use my HP-16c & DM16 calculators all the time. The programs here augment the 42's capabilities with every function on the 16c that I use. Functionality beyond the 16c includes comprehensive support for bit fields, better masks, integer square root & logarithms, and more. Finally this software provides a `SHOW`-like function called `BVIEW` is capable of displaying a full 64-bit binary number on a standard 42s display.

Everything is encapsulated into a single program with multiple global labels so that most of the functionality can be called directly from other programs.

The only thing that still bugs me is the annoying way the 42s requires one to enter hexadecimal digits. Not much I can do about that...

3 `BASE`: BASE-N Application

3.1 Functionality & Menu

Menu:SubMenu:Key	Program	Description
S&R:SLn		Shift Y Left X Bits
S&R:SRn		Shift Y Right X Bits
S&R:RLn		Rotate Y Left by X bits
S&R:RRn		Rotate Y Right by X bits
S&R:□□□□		
S&R:BVIEW		
S&R:LJ		Left Justify Bits.
S&R:RJ		Right Justify Bits
S&R:SHFXY		Shift X by Y Bits (left when Y negative)
S&R:ROTXY		
S&R:□□□□		
S&R:BVIEW		

Continued on next page

Continued from previous page

Menu:SubMenu:Key	Program	Description
S&R:SL		Shift Left 1 Bit
S&R:SR		Shift Right 1 Bit
S&R:RL		Rotate Left 1 Bit
S&R:RR		Rotate Right 1 Bit
S&R:□□□□		
S&R:BVIEW		
BITS:GETB		Return the Xth bit of Y
BITS:SB	SETB	Set the Xth bit of Y
BITS:CB	CLRB	Clear the Xth bit in Y
BITS:AB	ASNB	Assign X to the Yth bit of Z
BITS:□□□□		
BITS:BVIEW		
BITS:GY@X		Return the Y bits of Z starting at bit X
BITS:SY@X		Set Y bits of Z starting at bit X
BITS:CY@X		Clear Y bits of Z starting at bit X
BITS:AZ@Y		Assign LS Z bits of X to Z bits of T starting at Y
BITS:□□□□		
BITS:BVIEW		
BITS:POPRB		Pop rightmost bits X bits off of Y
BITS:POPLB		Pop leftmost bits X bits off of Y
BITS:□□□□		
BITS:□□□□		
BITS:□□□□		
BITS:BVIEW		
FUN:B#		Count set bits
FUN:MSSB		Position of the Most Significant Set Bit
FUN:LSSB		Position of the Least Significant Set Bit
FUN:ILN2		Integer base 2 log
FUN:ISQRT		Integer square root
FUN:BVIEW		
FUN:REVNB		Reverse rightmost X bits of Y
FUN:REVB		Reverse bits of X
FUN:□□□□		
FUN:□□□□		
FUN:□□□□		
FUN:BVIEW		
MASK:MSKL		Create integer with left most X bits set
MASK:MSKR		Create integer with right most X bits set
MASK:MSKn		Create integer with Y set bits located at bit X
MASK:□□□□		
MASK:□□□□		
MASK:BVIEW		
ARTH:AND		
ARTH:OR		
ARTH:XOR		
ARTH:NOT		
ARTH:NEG	BASE+/-	
ARTH:BVIEW		
ARTH:+	BASE+	
ARTH:-	BASE-	
ARTH:×	BASE×	
ARTH:÷	BASE÷	
ARTH:MOD		
ARTH:BVIEW		
BVIEW		
LBL 92	HEXM	Set current base to hexadecimal
LBL 93	DECM	Set current base to decimal
LBL 94	OCTM	Set current base to octal
LBL 95	BINM	Set current base to binary
□□□□	-	
BMNU	LBL 75	Switch to built in BASE menu
LBL 96	WSIZE	Set WSIZE
LBL 97	BSIGNED	Toggle signed/unsigned mode
LBL 98	BWRAP	Toggle wrapped mode
□□□□	-	
LBL 99	LBL 74	Toggle BVIEW padding
□□□□	-	

3.2 Menu Code

The menu program is generated via the following bit of elisp. You must first define the `MJR-generate-42-menu-code` and `MJR-custom-x-gen` by evaluating the code blocks in the `hp42s-meta.org` file. The skeleton was generated by the following code. That skeleton has been fleshed out with some custom code.

```
(MJR-generate-42-menu-code "BASE" 0 tbl 0 1 'stay 'up 'auto #'MJR-custom-gen-lab #'MJR-custom-gen-sub)

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@      (BASE)
@@@@ DSC: Auto-generated menu program
LBL "BASE"
LBL 01              @@@@ Page 1 of menu BASE
CLMENU
"S&R"
KEY 1 GTO 04
"BITS"
KEY 2 GTO 05
"FUN"
KEY 3 GTO 06
"MASK"
KEY 4 GTO 07
"ARTH"
KEY 5 GTO 08
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 03
KEY 8 GTO 02
KEY 9 GTO 00
MENU
STOP
GTO 01
LBL 02              @@@@ Page 2 of menu BASE
CLMENU
XEQ 92
KEY 1 XEQ 09
XEQ 93
KEY 2 XEQ 10
XEQ 94
KEY 3 XEQ 11
XEQ 95
KEY 4 XEQ 12
"BMNU"
KEY 6 XEQ 75
KEY 7 GTO 01
KEY 8 GTO 03
KEY 9 GTO 00
MENU
STOP
GTO 02
LBL 03              @@@@ Page 3 of menu BASE
CLMENU
XEQ 96
KEY 1 XEQ 13
XEQ 97
KEY 2 XEQ 14
XEQ 98
KEY 3 XEQ 15
XEQ 99
KEY 5 XEQ 74
KEY 7 GTO 02
KEY 8 GTO 01
KEY 9 GTO 00
MENU
STOP
GTO 03
LBL 04              @@@@ Page 1 of menu S&R
CLMENU
"SLn"
KEY 1 XEQ "SLn"
"SRn"
KEY 2 XEQ "SRn"
"RLn"
KEY 3 XEQ "RLn"
"RRn"
KEY 4 XEQ "RRn"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 17
KEY 8 GTO 16
KEY 9 GTO 01
MENU
STOP
```

```
GTO 04
LBL 16            @@@@ Page 2 of menu S&R
CLMENU
"LJ"
KEY 1 XEQ "LJ"
"RJ"
KEY 2 XEQ "RJ"
"SHFXY"
KEY 3 XEQ "SHFXY"
"ROTXY"
KEY 4 XEQ 18
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 04
KEY 8 GTO 17
KEY 9 GTO 01
MENU
STOP
GTO 16
LBL 17            @@@@ Page 3 of menu S&R
CLMENU
"SL"
KEY 1 XEQ "SL"
"SR"
KEY 2 XEQ "SR"
"RL"
KEY 3 XEQ "RL"
"RR"
KEY 4 XEQ "RR"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 16
KEY 8 GTO 04
KEY 9 GTO 01
MENU
STOP
GTO 17
LBL 05            @@@@ Page 1 of menu BITS
CLMENU
"GETB"
KEY 1 XEQ "GETB"
"SB"
KEY 2 XEQ "SETB"
"CB"
KEY 3 XEQ "CLRB"
"AB"
KEY 4 XEQ "ASNB"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 20
KEY 8 GTO 19
KEY 9 GTO 01
MENU
STOP
GTO 05
LBL 19            @@@@ Page 2 of menu BITS
CLMENU
"GY@X"
KEY 1 XEQ "GY@X"
"SY@X"
KEY 2 XEQ "SY@X"
"CY@X"
KEY 3 XEQ "CY@X"
"AZ@Y"
KEY 4 XEQ "AZ@Y"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 05
KEY 8 GTO 20
KEY 9 GTO 01
MENU
STOP
GTO 19
LBL 20            @@@@ Page 3 of menu BITS
CLMENU
"POPRB"
```

```
KEY 1 XEQ "POPRB"
"POPLB"
KEY 2 XEQ "POPLB"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 19
KEY 8 GTO 05
KEY 9 GTO 01
MENU
STOP
GTO 20
LBL 06            @@@@ Page 1 of menu FUN
CLMENU
"B#"
KEY 1 XEQ "B#"
"MSSB"
KEY 2 XEQ "MSSB"
"LSSB"
KEY 3 XEQ "LSSB"
"ILN2"
KEY 4 XEQ "ILN2"
"ISQRT"
KEY 5 XEQ "ISQRT"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 21
KEY 8 GTO 21
KEY 9 GTO 01
MENU
STOP
GTO 06
LBL 21            @@@@ Page 2 of menu FUN
CLMENU
"REVNB"
KEY 1 XEQ "REVNB"
"REVB"
KEY 2 XEQ "REVB"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 06
KEY 8 GTO 06
KEY 9 GTO 01
MENU
STOP
GTO 21
LBL 07            @@@@ Page 1 of menu MASK
CLMENU
"MSKL"
KEY 1 XEQ "MSKL"
"MSKR"
KEY 2 XEQ "MSKR"
"MSKn"
KEY 3 XEQ "MSKn"
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 9 GTO 01
MENU
STOP
GTO 07
LBL 08            @@@@ Page 1 of menu ARTH
CLMENU
"AND"
KEY 1 XEQ 23
"OR"
KEY 2 XEQ 24
"XOR"
KEY 3 XEQ 25
"NOT"
KEY 4 XEQ 26
"NEG"
KEY 5 XEQ 27
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 22
KEY 8 GTO 22
KEY 9 GTO 01
```

```
MENU
STOP
GTO 08
LBL 22            @@@@ Page 2 of menu ARTH
CLMENU
"+"
KEY 1 XEQ 28
"-"
KEY 2 XEQ 29
"×"
KEY 3 XEQ 30
"÷"
KEY 4 XEQ 31
"MOD"
KEY 5 XEQ 32
"BVIEW"
KEY 6 XEQ "BVIEW"
KEY 7 GTO 08
KEY 8 GTO 08
KEY 9 GTO 01
MENU
STOP
GTO 22
LBL 00 @@@@ Application Exit
EXITALL
RTN
LBL 09    @@@@ Action for menu key LBL 92
HEXM
RTN
LBL 10    @@@@ Action for menu key LBL 93
DECM
RTN
LBL 11    @@@@ Action for menu key LBL 94
OCTM
RTN
LBL 12    @@@@ Action for menu key LBL 95
BINM
RTN
LBL 13    @@@@ Action for menu key LBL 96
WSIZE
RTN
LBL 14    @@@@ Action for menu key LBL 97
BSIGNED
RTN
LBL 15    @@@@ Action for menu key LBL 98
BWRAP
RTN
LBL 18    @@@@ Action for menu key ROTXY
ROTXY
RTN
LBL 23    @@@@ Action for menu key AND
AND
RTN
LBL 24    @@@@ Action for menu key OR
OR
RTN
LBL 25    @@@@ Action for menu key XOR
XOR
RTN
LBL 26    @@@@ Action for menu key NOT
NOT
RTN
LBL 27    @@@@ Action for menu key NEG
BASE+/-
RTN
LBL 28    @@@@ Action for menu key +
BASE+
RTN
LBL 29    @@@@ Action for menu key -
BASE-
RTN
LBL 30    @@@@ Action for menu key ×
BASE×
RTN
LBL 31    @@@@ Action for menu key ÷
BASE÷
```

```
RTN
LBL 32    @@@@ Action for menu key MOD
MOD
RTN
@@@@ Free labels start at: 33
```

3.3 Application Local Subroutines

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ (ref:b#)
@@@@ DSC: Number of bits set
@@@@ IN:  X: an integer
@@@@ OUT: X: Number of 1 bits in IP(X)
@@@@ FAQ: Runtime complexity O(log_2(X))
@@@@ LBL: Used 70-72
@@@@ UPD: 2021-04-10
@@@@ LBL "B#"
LBL "B#"
FUNC 11    @@## REQ:free42>=2.5.24
L4STK      @@## REQ:free42>=3.0
IP
0
X<>Y       @@@@ NUM CNT
1
NOT
X<>Y       @@@@ NUM MASK CNT
LBL 70
RCL ST Y   @@@@ MASK NUM MASK CNT
X<>Y       @@@@ NUM MASK MASK CNT
AND        @@@@ NUM_N MASK CNT
LASTX      @@@@ NUM NUM_N MASK CNT
X=Y?
GTO 71
@@@@ Current bit was set: increment counter
R↓         @@@@ NUM_N MASK CNT
1
STO+ ST T  @@@@ 1 NUM_N MASK CNT
LBL 71
@@@@ Current bit was clear
R↓         @@@@ NUM_N MASK CNT
X=0?
GTO 72
@@@@ Still have bits to check
X<>Y       @@@@ MASK NUM_N CNT
-1
ROTXY
X<>Y       @@@@ NUM_N MASK CNT
GTO 70
LBL 72
@@@@ No bits left to check
R↓         @@@@ MASK CNT
R↓         @@@@ CNT
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (MSKn)
@@@@ DSC: Create integer with Y set bits located at bit X
@@@@ IN:  Y: An integer
@@@@ IN:  X: An integer (LSB=0)
@@@@ OUT: X: Integer with IP(X) least significant bits set
@@@@ FAQ: Returns int with all 0 bits when X<=0
@@@@ FAQ: Returns int with all 1 bits when X>=WSIZE?
@@@@ UPD: 2021-03-20
LBL "MSKn"
FUNC 21                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
X<>Y
XEQ "MSKR"
X<>Y
XEQ "SLn"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (MSKL)
@@@@ DSC: Create integer with left most X bits set
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer with IP(X) most significant bits set
@@@@ FAQ: Returns int with all 0 bits when X<=0
@@@@ FAQ: Returns int with all 1 bits when X>=WSIZE?
```

```
@@@@ UPD: 2021-03-20
LBL "MSKL"
FUNC 11                         @@## REQ:free42>=2.5.24
L4STK                           @@## REQ:free42>=3.0
XEQ "MSKR"
XEQ "LJ"
R↓
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (MSKR)
@@@@ DSC: Create integer with right most X bits set
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer with IP(X) least significant bits set
@@@@ FAQ: Returns int with all 0 bits when X<=0
@@@@ FAQ: Returns int with all 1 bits when X>=WSIZE?
@@@@ UPD: 2021-03-20
LBL "MSKR"
FUNC 11                         @@## REQ:free42>=2.5.24
L4STK                           @@## REQ:free42>=3.0
IP
@@@@ All zeros case?
0
0≥? ST Y            @@## REQ:free42>=3.0.3
XEQ 44 @@@ 0≥Y?     @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
R↓
0
NOT
@@@@ All ones case?
XEQ 67 @@@@ Y≥WSIZE?
RTN
R↓
@@@@ Some ones case
2
X<>Y
Y↑X
1
-
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (SETB)
@@@@ DSC: Set the Xth bit of Y
@@@@ IN:  Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: X: Integer with IP(X)th bit of Y set
@@@@ FAQ: Y is returned unchanged  when X is out of range
@@@@ BUG: When X out of range, the type Y is not error checked
@@@@ UPD: 2021-03-20
LBL "SETB"
FUNC 21                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
IP
X<>Y
@@@@ Do nothing if X was negative
0>? ST Y           @@## REQ:free42>=3.0.3
XEQ 43 @@@ 0>Y?    @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
@@@@ Do nothing if X was >= WSIZE
XEQ 67 @@@@ Y≥WSIZE?
RTN
@@@@ X must have been >= 0 & < WSIZE
X<>Y
1
X<>Y
+/-
ROTXY
OR
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (CLRB)
@@@@ DSC: Clear the Xth bit in Y
@@@@ IN:  Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: X: Integer with IP(X)th bit of Y set
@@@@ FAQ: Y is returned unchanged  when X is out of range
@@@@ BUG: When X out of range, the type Y is not error checked
```

```
@@@@ UPD: 2021-03-20
LBL "CLRB"
FUNC 21                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
IP
X<>Y
IP
@@@@ Do nothing if X was negative
0>? ST Y           @@## REQ:free42>=3.0.3
XEQ 43 @@@ 0>Y?    @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
@@@@ Do nothing if X was >= WSIZE
XEQ 67 @@@@ Y⩾WSIZE?
RTN
@@@@ X must have been >= 0 & < WSIZE
X<>Y
1
X<>Y
+/-
ROTXY
NOT
AND
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (GETB)
@@@@ DSC: Return the Xth bit of Y
@@@@ IN:  Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: The IP(X)th bit of Y
@@@@ FAQ: Returns 0 when X is out of range
@@@@ BUG: When X out of range, type Y is not error checked
@@@@ LBL: Used 69
@@@@ UPD: 2021-03-20
LBL "GETB"
FUNC 21                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
0
@@@@ Return 0 if X was negative
0>? ST Y           @@## REQ:free42>=3.0.3
XEQ 43 @@@ 0>Y?    @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
@@@@ Return 0  if X was >= WSIZE
XEQ 67 @@@@ Y⩾WSIZE?
RTN
@@@@ X must have been >= 0 & < WSIZE
R↓
BIT?
GTO 69
0
RTN
LBL 69
1
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (ASNB)
@@@@ DSC: Set the Yth bit of Z to X
@@@@ IN:  Z: An integer
@@@@ IN:  Y: An integer (0=LSB)
@@@@      X: An integer
@@@@ OUT: Set the IP(Y)th bit of IP(Z) to IP(X)
@@@@ UPD: 2021-03-20
LBL "ASNB"
FUNC 31                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
X=0?
GTO 54
R↓
XEQ "SETB"
RTN
LBL 54
R↓
XEQ "CLRB"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (GYaX)
@@@@ DSC: Return the Y bits of Z starting at bit X
```

```
@@@@ EX:  Z: 111000 Y: 3 X: 2 -> X: 110
@@@@ IN:  Z: An integer
@@@@      Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: IP(Y) bits of IP(Z) starting at bit IP(X)
@@@@ UPD: 2021-04-21
LBL "GY@X"
FUNC 31                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
RCL ST Z   @@@@ Z     X     Y     Z
X<>Y       @@@@ X     Z     Y     Z
XEQ "SRn"  @@@@ SHF_Z Y     Z     Z
X<>Y       @@@@ Y     SHF_Z Z     Z
XEQ "MSKR" @@@@ MASK  SHF_Z Z     Z
AND        @@@@ BITS  Z     Z     Z
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@                (CYaX)
@@@@ DSC: Clear Y bits of Z starting at bit X
@@@@ EX:  Z: 111000 Y: 3 X: 2 -> X: 100000
@@@@ IN:  Z: An integer
@@@@      Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: IP(Z) with IP(Y) bits starting at bit IP(X) cleared
@@@@ UPD: 2021-04-21
LBL "CY@X"
FUNC 31                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
XEQ "MSKn"
NOT
AND
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@                (SYaX)
@@@@ DSC: Set Y bits of Z starting at bit X
@@@@ EX:  Z: 111000 Y: 3 X: 2 -> X: 111100
@@@@ IN:  Z: An integer
@@@@      Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: IP(Z) with IP(Y) bits starting at bit IP(X) set
@@@@ UPD: 2021-04-21
LBL "SY@X"
FUNC 31                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
XEQ "MSKn"
OR
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@                (AZaY)
@@@@ DSC: Place Z bits of X into T starting at bit Y
@@@@ EX:  T: 111000 Z: 3 Y: 2 X: 011 -> X: 101100
@@@@ IN:  T: An integer
@@@@ IN:  Z: An integer
@@@@      Y: An integer
@@@@      X: An integer (0=LSB)
@@@@ OUT: IP(Z) bits of IP(T) are set to IP(Z) bits of IP(X) starting at IP(Y)
@@@@ UPD: 2021-04-21
LBL "AZ@Y"
FUNC 41                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
LSTO "_X"
R↓              @@@@  Y Z T T
LSTO "_Y"
R↓              @@@@  Z T T T
LSTO "_Z"     @@@@  Z T T T
RCL "_Y"      @@@@  Y Z T T
XEQ "MSKn"
NOT
AND             @@@@ TM T T T
RCL "_X"
RCL "_Z"
XEQ "MSKR"
AND
RCL "_Y"
XEQ "SLn"
OR              @@@@
```

```
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (SL)
@@@@ DSC: Shift Left 1 Bit
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer shifted left 1 bit
@@@@ UPD: 2021-03-20
LBL "SL"
FUNC 11                     @@## REQ:free42>=2.5.24
L4STK                       @@## REQ:free42>=3.0
-1
ROTXY
1
NOT
AND
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (SR)
@@@@ DSC: Shift Right 1 Bit
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer shifted right 1 bit
@@@@ UPD: 2021-03-20
LBL "SR"
FUNC 11                     @@## REQ:free42>=2.5.24
L4STK                       @@## REQ:free42>=3.0
1
NOT
AND
1
ROTXY
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (RL)
@@@@ DSC: Rotate Left 1 Bit
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer rotated left 1 bit
LBL "RL"
FUNC 11                     @@## REQ:free42>=2.5.24
L4STK                       @@## REQ:free42>=3.0
-1
ROTXY
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (RR)
@@@@ DSC: Rotate Right 1 Bit
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer rotated right 1 bit
LBL "RR"
FUNC 11                     @@## REQ:free42>=2.5.24
L4STK                       @@## REQ:free42>=3.0
1
ROTXY
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (SLn)
@@@@ DSC: Shift Y Left X Bits
@@@@ IN:  Y: An integer
@@@@ IN:  X: An integer
@@@@ OUT: X: IP(Y) shifted left IP(Y) bit(s)
@@@@ FAQ: IP(Y) is returned unchanged when IP(X) is zero or negative
@@@@ FAQ: 0 is returned when IP(X)>=WSIZE
@@@@ UPD: 2021-03-20
LBL "SLn"
FUNC 21                     @@## REQ:free42>=2.5.24
L4STK                       @@## REQ:free42>=3.0
IP
X<>Y
IP
@@@@ Return Y if X was negative
0≥? ST Y          @@## REQ:free42>=3.0.3
XEQ 44 @@@ 0≥Y?    @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
X<>Y
0
@@@@ Return 0 if X was >= WSIZE
XEQ 67 @@@@ Y≥WSIZE?
```

```
RTN
@@@@ X must have been > 0 & < WSIZE
R↓
X<>Y
RCL ST Y
@@@@ Rotate left X bits
+/-
ROTXY
@@@@ Zap the right most X bits
2
RCL ST Z
Y↑X
1
-
NOT
AND
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (SRn)
@@@@ DSC: Shift Y Right X Bits
@@@@ IN:  Y: An integer
@@@@ IN:  X: An integer
@@@@ OUT: X: IP(Y) shifted right IP(X) bit(s)
@@@@ FAQ: IP(Y) is returned unchanged when IP(X) is zero or negative
@@@@ FAQ: 0 is returned when IP(X)>=WSIZE
@@@@ UPD: 2021-03-20
LBL "SRn"
FUNC 21                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
IP
X<>Y
IP
@@@@ Return Y if X was negative
0≥? ST Y           @@## REQ:free42>=3.0.3
XEQ 44 @@@ 0≥Y?    @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
X<>Y
0
@@@@ Return 0 if X was >= WSIZE
XEQ 67 @@@@ Y≥WSIZE?
RTN
@@@@ X must have been > 0 & < WSIZE
R↓
@@@@ Zap the right most X bits
X<>Y
RCL ST Y
2
X<>Y
Y↑X
1
-
NOT
AND
@@@@ Rotate right X bits
X<>Y
ROTXY
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (RLn)
@@@@ DSC: Rotate Y Left by X bits
@@@@ IN:  Y: An integer
@@@@ IN:  X: An integer
@@@@ OUT: X: X rotated left IP(Y) bit(s)
@@@@ FAQ: Y is returned unchanged when X is zero or negative
@@@@ UPD: 2021-03-20
LBL "RLn"
FUNC 21                       @@## REQ:free42>=2.5.24
L4STK                         @@## REQ:free42>=3.0
IP
X<>Y
@@@@ Return Y if X was negative
0≥? ST Y           @@## REQ:free42>=3.0.3
XEQ 44 @@@ 0≥Y?    @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
X<>Y
@@@@ X was positive
```

```
+/-
ROTXY
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (RRn)
@@@@ DSC: Rotate Y Right by X bits
@@@@ IN:  Y: An integer
@@@@ IN:  X: An integer
@@@@ OUT: X: X rotated right ABS(IP(Y)) bit(s)
@@@@ FAQ: Y is returned unchanged when X is zero or negative
@@@@ UPD: 2021-03-20
LBL "RRn"
FUNC 21                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
IP
X<>Y
@@@@ Return Y if X was negative
0≥? ST Y            @@## REQ:free42>=3.0.3
XEQ 44 @@@ 0≥Y?     @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
X<>Y
@@@@ X was positive
ROTXY
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (SHFXY)
@@@@ DSC: Shift X by Y Bits (left when Y negative)
@@@@ IN:  Y: An integer
@@@@ IN:  X: An integer
@@@@ OUT: X: Integer shifted by IP(Y) bit(s)
@@@@ FAQ: Uses SLn & SLr.
@@@@ LBL: Used 59
@@@@ UPD: 2021-03-20
LBL "SHFXY"
FUNC 21                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
IP
X<0?
GTO 59
@@@@ X is non-negative -- shift right
XEQ "SRn"
RTN
LBL 59
@@@@ X negative -- shift left
+/-
XEQ "SLn"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (RJ)
@@@@ DSC: Right Justify Bits
@@@@ IN:  X: an integer
@@@@ OUT: Y: Number of shifts required to justify
@@@@ OUT: X: Justified number
@@@@ FAQ: Like the HP-16c's LJ funciton, but justifies in the other direction
@@@@ LBL: Used 56-58
@@@@ UPD: 2021-03-20
LBL "RJ"
FUNC 12                         @@## REQ:free42>=2.5.24
L4STK                           @@## REQ:free42>=3.0
IP
0
X<>Y
LBL 57
0
BIT?
GTO 56
GTO 58
LBL 56   @@@@ LSB is one -- DONE!
R↓
X<>Y
RTN
LBL 58   @@@@ LSB is zero
R↓
1
STO+ ST Z
ROTXY
```

```
GTO 57

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (LJ)
@@@@ DSC: Left Justify Bits
@@@@ IN:  X: an integer
@@@@ OUT: Y: Number of shifts required to justify
@@@@ OUT: X: Justified number
@@@@ FAQ: Just like the HP-16c's LJ funciton
@@@@ LBL: Used 64-66
@@@@ UPD: 2021-03-20
LBL "LJ"
FUNC 12                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
IP
0
X<>Y
LBL 64
WSIZE?
1
-
BIT?
GTO 65
GTO 66
LBL 65   @@@@ MSB is one -- DONE!
R↓
X<>Y
RTN
LBL 66   @@@@ MSB is zero
R↓
1
STO+ ST Z
+/-
ROTXY
GTO 64

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (ISQRT)
@@@@ DSC: Integer square root
@@@@ IN:  X: a non-negative real number
@@@@ OUT: X: IP(SQRT(ABS(X)))
@@@@ UPD: 2021-03-16
LBL "ISQRT"
FUNC 11                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
ABS
SQRT
IP
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (ILN2)
@@@@ DSC: Integer base 2 log
@@@@ IN:  X: a positive real number
@@@@ OUT: X: IP(log_2(ABS(X)))
@@@@ UPD: 2021-03-16
LBL "ILN2"
FUNC 11                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
ABS
LN
2
LN
÷
IP
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (MSSB)
@@@@ DSC: Position of the Most Significant Set Bit
@@@@ IN:  X: An integer
@@@@ OUT: X: Position of MSSB in IP(X) or -1 if IP(X) was 0
@@@@ USE: LJ
@@@@ UPD: 2021-03-16
@@@@ FAQ: Runtime Complexity O(WSIZE-log_2(X))
@@@@ FAQ: Closely related to LJ program
LBL "MSSB"
FUNC 11                        @@## REQ:free42>=2.5.24
L4STK                          @@## REQ:free42>=3.0
IP
```

```
-1
0=? ST Y          @@## REQ:free42>=3.0.3
XEQ  42 @@@ 0=Y?  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
R↓
XEQ "LJ"
1
+
WSIZE?
X<>Y
-
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (LSSB)
@@@@ DSC: Position of the Least Significant Set Bit
@@@@ IN:  X: An integer
@@@@ OUT: X: Position of LSSB in IP(X) or -1 if IP(X) was 0
@@@@ UPD: 2021-03-16
@@@@ FAQ: Runtime Complexity O(LSSB(X)) & LSSB<WSIZE
@@@@ FAQ: Closely related to RJ program
LBL "LSSB"
FUNC 11                         @@## REQ:free42>=2.5.24
L4STK                           @@## REQ:free42>=3.0
IP
-1
0=? ST Y          @@## REQ:free42>=3.0.3
XEQ  42 @@@ 0=Y?  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
R↓
XEQ "RJ"
RTN
                      @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
LBL 42   @@@ 0=Y?     @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
FUNC 00               @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
X<>Y                  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
X=0?                  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTNYES                @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTNNO                 @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
LBL 43   @@@ 0>Y?     @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
FUNC 00               @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
X<>Y                  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
X<0?                  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTNYES                @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTNNO                 @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
LBL 44   @@@ 0≥Y?     @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
FUNC 00               @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
X<>Y                  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
X≤0?                  @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTNYES                @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
RTNNO                 @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
                      @@## REQ:free42<3.0.3   @@## TODO: Delete when DM42 >= free42 3.0.3
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: WSIZE
LBL 96
"W:"
WSIZE?
AIP
R↓
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: BSIGNED
LBL 97
FS? 78
"S:+/-"
FC? 78
"S:+"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: BWRAP
LBL 98
FS? 79
"BWR•"
FC? 79
"BWRP"
```

```
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: BVIEW
LBL 99
FC? 02
"BVA•"
FS? 02
"BVA"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: HEX
LBL 92
"HEX"
FS? 71
⊢"•"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: DEC
LBL 93
"DEC"
SF 81
FS? 71
CF 81
FS? 70
CF 81
FS? 68
CF 81
FS? 81
⊢"•"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: OCT
LBL 94
"OCT"
SF 81
FS? 71
CF 81
FC? 70
CF 81
FS? 81
⊢"•"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Label: BIN
LBL 95
"BIN"
SF 81
FC? 68
CF 81
FS? 69
CF 81
FS? 81
⊢"•"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Toggle 02
LBL 74
FS?C 02
RTN
SF 02
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ Menu Action BINM
LBL 75
EXITALL
"Press R/S To"
"⊢ Return"
AVIEW
HEXM
```

```
STOP
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ DSC: Test if Y<WSIZE
@@@@ NAM: Y⩾WSIZE? 67
@@@@ I/O: No stack change.  Uses Y
@@@@ RET: YES if Y⩾WSIZE, NO otherwise
@@@@ UPD: 2021-04-21
LBL 67
FUNC 00
L4STK
X<>Y
WSIZE?
X>Y?
RTNNO
RTNYES

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (REVBIT)
@@@@ DSC: Reverse Bits rightmost X bits of Y
@@@@ NAM: REVBIT
@@@@ IN:  Y: An integer
@@@@      X: An integer
@@@@ OUT: X: An integer
@@@@ UPD: 2021-04-21
LBL "REVNB"
FUNC 11
L4STK
ENTER           @@@@ X X Y
XEQ "MSKR"      @@@@ M X Y
X<>Y            @@@@ X M Y
1
-
LSTO "_MBC"
1000
÷
LSTO "_CTR"
R↓              @@@@ M Y
RCL ST Y        @@@@ Y M Y
OR              @@@@ N Y
X<>Y            @@@@ OLD NEW
LBL 51
RCL "_CTR"       @@@@ CTR OLD NEW
IP
BIT?
GTO 52
R↓
X<>Y
RCL "_CTR"
IP
RCL- "_MBC"
1
X<>Y
ROTXY
NOT
AND
X<>Y
ENTER
LBL 52
R↓
ISG "_CTR"
GTO 51
X<>Y
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@               (REVBIT)
@@@@ DSC: Reverse all Bits of X
@@@@ NAM: REVBIT
@@@@ IN:  X: An integer
@@@@ OUT: X: An integer
@@@@ UPD: 2021-04-21
LBL "REVB"
FUNC 11
L4STK
WSIZE?
XEQ "REVNB"
```

```
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (POPRB)
@@@@ DSC: Shift rihtmost X bits off Y.
@@@@ IN:  Y: An integer
@@@@      X: An integer (number of bits)
@@@@ OUT: Y: Part of Y left
@@@@      X: Part of Y shifted off
LBL "POPRB"
FUNC 22
L4STK
RCL ST Y     @@@@ Y   X   Y   ?
RCL ST Y     @@@@ X   Y   X   Y
XEQ "SRn"    @@@@ LFT X   Y   Y
RCL ST Z     @@@@ Y   LFT X   Y
RCL ST Z     @@@@ X   Y   LFT X
XEQ "MSKR"   @@@@ MSK Y   LFT X
AND          @@@@ POP LFT X   Y
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (POPLB)
@@@@ DSC: Shift leftmost X bits off Y.
@@@@ IN:  Y: An integer
@@@@      X: An integer (number of bits)
@@@@ OUT: Y: Part of Y left
@@@@      X: Part of Y shifted off
LBL "POPLB"
FUNC 22
L4STK
RCL ST Y     @@@@ Y   X   Y   ?
RCL ST Y     @@@@ X   Y   X   Y
XEQ "SLn"    @@@@ LFT X   Y   Y
RCL ST Z     @@@@ Y   LFT X   Y
RCL ST Z     @@@@ X   Y   LFT X
XEQ "MSKL"   @@@@ MSK Y   LFT X
AND          @@@@ PS  LFT X   Y
RCL ST Z
+/-
ROTXY        @@@@ POP LFT X   Y
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@          (BVIEW)
@@@@ DSC: Print binary numbers (up to 64-bit)
@@@@ IN:  X: An integer
@@@@ OUT: Nothing -- prints to screen
@@@@ FLG: 81: TEMP: Set: We are on DM42;            Clear: We are not on DM42
@@@@ FLG: 01: PREF: Reserved
@@@@ FLG: 02: PREF: Set: BVIEW prints WSIZE digits; Clear: BVIEW prints 64 digits
@@@@ FAQ: If XEQ while system BASE menu is active, returns to same menu.
@@@@ BUG: Sometimes gets the system base menu to return to wrong. ;)
@@@@ UPD: 2021-03-20
LBL "BVIEW"
FUNC 11              @@## REQ:free42>=2.5.24
L4STK                @@## REQ:free42>=3.0
0
SF 25
BIT?
FS?C 25
GTO 79
R↓
"ERR: Bad Int!"
AVIEW
RTN
LBL 79
R↓
LSTO "_X"
20
1
FS? 68
STO+ ST Y
FS? 70
STO+ ST Y
FS? 71
STO+ ST Y
R↓              @@@@ 20:DECN 21:BINM 22:OCTM 23:HEXM
LSTO "_SBMD"
```

```
R↓
CF 81
SF 25
RCL "GrMod"
FS?C 25
SF 81
FC? 81
GTO 88
@@@@ We are on a DM42
0
STO "GrMod"     @@@@ Set graphics mode to 42 classic on DM42
R↓
LBL 88
CLLCD
EXITALL
63
1000
÷
LSTO "_CTR"
LBL 91
LBL 76  @@@@ TODO.  Not used?
@@@@ Figure out X&Y coordinates for digit
RCL "_CTR"            @@@@ IF
IP
32
X>Y?
GTO 73
GTO 83
LBL 73                @@@@ IF-THEN FIRST ROW
1
RCL "_CTR"
4
×
1
+
GTO 90
LBL 83                @@@@ IF-ELSE SECOND ROW
9
RCL "_CTR"
32
-
4
×
1
+
LBL 90                @@@@ IF-END
@@@@ Figure out current bit value
63
RCL "_CTR"            @@@@ IF-BEGIN
IP
-
WSIZE?
X>Y?
GTO 80
GTO 81
LBL 80                @@@@ IF-THEN
R↓
+/-                   @@@@ IF-BEGIN
1
X<>Y
ROTXY
RCL "_X"
AND
X=0?
GTO 85
GTO 86
LBL 85                @@@@ IF-THEN bit is 0
R↓
0
GTO 87
LBL 86                @@@@ IF-ELSE bit is 1
R↓
1
LBL 87
LSTO "_CB"
GTO 82
```

```
LBL 81              @@@@ IF-ELSE
R↓
R↓
@@@@ Bit beyond WSIZE padding character
0
LSTO "_CB"
FS? 02
GTO 84
LBL 82              @@@@ IF-END
R↓
@@@@ Figure out grouping
RCL "_CTR"           @@@@ IF-START grouping
IP
4
÷
IP
2
÷
FP
X=0?
GTO 55
GTO 68
LBL 55              @@@@ IF-THEN
R↓
0
GTO 77
LBL 68              @@@@ IF-ELSE
R↓
2
LBL 77              @@@@ IF-END
@@@@ Compute digit character code
RCL+ "_CB"
@@@@ Draw it
XEQ 78
LBL 84
ISG  "_CTR"
GTO 91
FC? 81
GTO 89
@@@@ On DM42.  Hack to keep screen clean
GETKEY
CLLCD
"BVIEW FINISHED"
AVIEW
LBL 89
RCL "_SBMD"
20
X=Y?
DECM
R↓
21
X=Y?
BINM
R↓
22
X=Y?
OCTM
R↓
23
X=Y?
HEXM
R↓
@@@@ Recall original X
RCL "_X"
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@ DSC: tiny binary digit print
@@@@ IN:  Z: Y coordinate for upper left point of character -- Top of screen is 1
@@@@      Y: X coordinate for upper left point of character -- Left of screen is 1
@@@@      X: Character number
@@@@ OUT: No return
@@@@ UPD: 2021-04-11
@@@@ BUG: Characters can be *VERY* tiny in high resolution modes on DM42
@@@@ Characters can be underlined
@@@@ Characters are 3x5 pixels in size.  Underlined characters are 3x7.
```

```
@@@@   - Stock HP-42s screen: 32 characters across.  Two full lines on the screen.
@@@@ Non-Underlined Character numbers:
@@@@     00 01
@@@@      0  1
@@@@ Add 2 to the above character number for the underlined version
LBL 78
FUNC 30      @@## REQ:free42>=2.5.24
L4STK        @@## REQ:free42>=3.0
IP
60
+
XEQ IND ST X
R↓
AGRAPH
RTN
LBL 60       @@@@ CHAR: 0
"•μ•"        @@@@ #b11111 #b10001 #b11111
RTN
LBL 61       @@@@ CHAR: 1
"£•←"        @@@@ #b10010 #b11111 #b10000
RTN
LBL 62       @@@@ CHAR: 0
"_Q_"        @@@@ #b1011111 #b1010001 #b1011111
RTN
LBL 63       @@@@ CHAR: 1
"R_P"        @@@@ #b1010010 #b1011111 #b1010000
RTN

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
END
```

4 EOF