### free42 Base-N Tools

## Mitch Richling

### 2021-03-19

Author: Mitch Richling Updated: 2021-05-13 19:55:18

Copyright 2021 Mitch Richling. All rights reserved.

#### Contents

1	Metadata	1
2	Introduction	1
	BASE: BASE-N Application 3.1 Functionality & Menu 3.2 Menu Code 3.3 Application Local Subroutines	2
4	FOF	21

### 1 Metadata

The home for this HTML file is:  $\verb|https://richmit.github.io/hp42/base.html|$ 

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:

docs - This html document

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
$S\&R:\Box\Box\Box\Box$		
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		
$S\&R:\Box\Box\Box\Box$		
S&R:BVIEW		

Continued on next page

Continued from previous Menu:SubMenu:Key	Program	Description
S&R:SL	1 Togram	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:		Totale Tugae I Bit
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
$BITS:\square\square\square\square$		
$BITS:\square\square\square\square$		
$BITS:\Box\Box\Box\Box$		
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
$MASK:\Box\Box\Box\Box$ $MASK:\Box\Box\Box\Box$		
MASK:BVIEW		
ARTH:AND		
ARTH:OR		
ARTH:XOR		
ARTH:NOT		
ARTH:NEG	BASE+/-	
ARTH:BVIEW	D1101   /-	
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
	_	33
LBL 99	LBL $74$	Toggle BVIEW padding
LDL 33		

# 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.

```
(BASE)
0000 DSC: Auto-generated menu program
LBL "BASE"
LBL 01
                0000 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
                0000 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
                 0000 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
                  0000 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"
```

4

```
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
                  @@@@ Page 1 of menu FUN
LBL 06
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
                  0000 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
                  0000 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
          0000 Action for menu key LBL 92
HEXM
RTN
LBL 10
          0000 Action for menu key LBL 93
DECM
RTN
LBL 11
          0000 Action for menu key LBL 94
OCTM
RTN
LBL 12
          0000 Action for menu key LBL 95
BINM
RTN
LBL 13
          0000 Action for menu key LBL 96
WSIZE
RTN
LBL 14
          0000 Action for menu key LBL 97
BSIGNED
RTN
          0000 Action for menu key LBL 98
LBL 15
BWRAP
RTN
          0000 Action for menu key ROTXY
LBL 18
ROTXY
RTN
LBL 23
          0000 Action for menu key AND
AND
RTN
LBL 24
          0000 Action for menu key OR
OR
RTN
LBL 25
          0000 Action for menu key XOR
XOR
RTN
LBL 26
          0000 Action for menu key NOT
NOT
RTN
LBL 27
          0000 Action for menu key NEG
BASE+/-
RTN
LBL 28
          0000 Action for menu key +
BASE+
RTN
LBL 29
          0000 Action for menu key -
BASE-
RTN
LBL 30
          0000 Action for menu key ×
BASE×
RTN
LBL 31
          0000 Action for menu key \div
BASE÷
```

```
RTN
LBL 32
        0000 Action for menu key MOD
MOD
RTN
0000 Free labels start at: 33
3.3 Application Local Subroutines
0000 DSC: Number of bits set
0000 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
0000 LBL "B#"
LBL "B#"
FUNC 11
         @@## REQ:free42>=2.5.24
L4STK
         @@## REQ:free42>=3.0
ΤP
0
X<>Y
         0000 NUM CNT
1
NOT
X<>Y
         @@@@ NUM MASK CNT
LBL 70
RCL ST Y @@@@ MASK NUM MASK CNT
         @@@@ NUM MASK MASK CNT
X<>Y
AND
         @@@@ NUM_N MASK CNT
         @@@@ NUM NUM_N MASK CNT
LASTX
X=Y?
GTO 71
0000 Current bit was set: increment counter
         @@@@ NUM_N MASK CNT
STO+ ST T @@@@ 1 NUM_N MASK CNT
LBL 71
0000 Current bit was clear
R↓
         @@@@ NUM_N MASK CNT
X=0?
GTO 72
0000 Still have bits to check
X<>Y
        @@@@ MASK NUM_N CNT
-1
ROTXY
X<>Y
         @@@@ NUM_N MASK CNT
GTO 70
LBL 72
0000 No bits left to check
R↓
       @@@@ MASK CNT
R↓
         0000 CNT
RTN
(MSKn)
0000 DSC: Create integer with Y set bits located at bit X
0000 IN: Y: An integer
@@@@ IN: X: An integer (LSB=0)
0000 OUT: X: Integer with IP(X) least significant bits set
0000 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
                          @@## REQ:free42>=3.0
L4STK
X<>Y
XEQ "MSKR"
X<>Y
XEQ "SLn"
RTN
                                                                                  (MSKL)
0000 DSC: Create integer with left most X bits set
0000 IN: X: An integer
0000 OUT: X: Integer with IP(X) most significant bits set
0000 FAQ: Returns int with all 0 bits when X\leq 0
0000 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)
0000 DSC: Create integer with right most X bits set
0000 IN: X: An integer
0000 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
I.4STK
                       @@## REQ:free42>=3.0
ΙP
0000 All zeros case?
0
0≽? ST Y
              @@## REQ:free42>=3.0.3
RTN
R↓
0
NOT
0000 All ones case?
XEQ 67 @@@@ Y≥WSIZE?
RTN
R↓
0000 Some ones case
2
X<>Y
Y↑X
1
R.TN
(SETB)
0000 DSC: Set the Xth bit of Y
0000 IN: Y: An integer
0000
       X: An integer (0=LSB)
0000 OUT: X: Integer with IP(X)th bit of Y set
0000 FAQ: Y is returned unchanged when X is out of range
0000 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
ΙP
X<>Y
0000 Do nothing if X was negative
0>? ST Y @@## REQ:free42>=3.0.3
0000 Do nothing if X was >= WSIZE
XEQ 67 @@@@ Y≥WSIZE?
R.TN
0000 X must have been >= 0 & < WSIZE
X<>Y
1
X<>Y
+/-
ROTXY
OR
RTN
                                                                          (CLRB)
0000 DSC: Clear the Xth bit in Y
0000 IN: Y: An integer
0000
       X: An integer (0=LSB)
0000 OUT: X: Integer with IP(X)th bit of Y set
0000 FAQ: Y is returned unchanged when X is out of range
0000 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
ΙP
X<>Y
ΙP
\tt @@@@ Do nothing if X was negative
RTN
@@@@ Do nothing if X was >= WSIZE
XEQ 67 @@@@ Y≥WSIZE?
R.TN
0000 X must have been >= 0 & < WSIZE
X<>Y
1
X<>Y
+/-
ROTXY
NOT
AND
RTN
(GETB)
@@@@ DSC: Return the Xth bit of Y
0000 IN: Y: An integer
0000
       X: An integer (0=LSB)
0000 OUT: The IP(X)th bit of Y
@@@@ FAQ: Returns O when X is out of range
\tt QQQQ \ 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
                       @@## REQ:free42>=3.0
L4STK
0000 Return O if X was negative
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
0000 X must have been >= 0 & < WSIZE
R.J
BIT?
GTO 69
RTN
LBL 69
RTN
(ASNB)
@@@@ DSC: Set the Yth bit of Z to X
0000 IN: Z: An integer
@@@@ IN: Y: An integer (0=LSB)
0000
       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)
```

0000 DSC: Return the Y bits of Z starting at bit X

```
@@@@ EX: Z: 111000 Y: 3 X: 2 -> X: 110
0000 IN: Z: An integer
             Y: An integer
0000
                 X: An integer (0=LSB)
0000 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 0000 Z X
                                                     Y Z
X<>Y 0000 Z X Y X 0000 X Z Y X X Y X X Y X X Y X X Y X X Y X X Y X X Y X X Y X X Y X X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X Y X
                                                                  Z
                                                                 7.
X<>Y @@@@ Y SHF_Z Z
XEQ "MSKR" @@@@ MASK SHF_Z Z
                                                                 Z
AND
               @@@@ BITS Z
RTN
(CYaX)
0000 DSC: Clear Y bits of Z starting at bit X
@@@@ EX: Z: 111000 Y: 3 X: 2 -> X: 100000
0000 IN: Z: An integer
0000 Y: An integer
                 X: An integer (0=LSB)
0000 OUT: IP(Z) with IP(Y) bits starting at bit IP(X) cleared
@@@@ UPD: 2021-04-21
LBL "CY@X"
FUNC 31
                                                         @@## REO: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
0000 IN: Z: An integer
0000
                 Y: An integer
0000
                   X: An integer (0=LSB)
0000 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.
(AZaY)
0000 DSC: Place Z bits of X into T starting at bit Y
@@@@ EX: T: 111000 Z: 3 Y: 2 X: 011 -> X: 101100
0000 IN: T: An integer
0000 IN: Z: An integer
0000
                   Y: An integer
രരരര
                   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
                                                          @@## REQ:free42>=3.0
L4STK
LSTO "_X"
R↓
                              0000 Y Z T T
LSTO "_Y"
                             0000 Z T T T
R↓
LSTO "_Z"
                           0000 Z T T T
RCL "_Y"
                           0000 Y Z T T
XEQ "MSKn"
NOT
AND
                              0000 TM T T T
RCL "_X"
RCL "_Z"
XEQ "MSKR"
AND
RCL "_Y"
XEQ "SLn"
                               0000
```

```
(SL)
@@@@ DSC: Shift Left 1 Bit
0000 IN: X: An integer
0000 OUT: X: Integer shifted left 1 bit
@@@@ UPD: 2021-03-20
LBL "SL"
                                                           @@## REQ:free42>=2.5.24
FUNC 11
L4STK
                                                           @@## REQ:free42>=3.0
ROTXY
1
NOT
AND
R.TN
(SR)
0000 DSC: Shift Right 1 Bit
0000 IN: X: An integer
0000 OUT: X: Integer shifted right 1 bit
@@@@ UPD: 2021-03-20
LBL "SR"
                                                          @@## REQ:free42>=2.5.24
FUNC 11
L4STK
                                                           @@## REQ:free42>=3.0
NOT
AND
1
ROTXY
RTN
occorrections of the contraction of the contracti
                                                                                                                                                                                             (RL)
@@@@ DSC: Rotate Left 1 Bit
0000 IN: X: An integer
0000 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
0000 IN: X: An integer
0000 OUT: X: Integer rotated right 1 bit
I.BI. "RR."
                                                           @@## REQ:free42>=2.5.24
FUNC 11
L4STK
                                                           @@## REQ:free42>=3.0
1
ROTXY
R.TN
(SLn)
@@@@ DSC: Shift Y Left X Bits
0000 IN: Y: An integer
0000 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
ΙP
X<>Y
ΙP
0000 Return Y if X was negative
                                   @@## REQ:free42>=3.0.3
0≽? ST Y
XEQ 44 @@@ 0≥Y?
                                      @@## REQ:free42<3.0.3
                                                                                     @@## TODO: Delete when DM42 >= free42 3.0.3
R.TN
X<>Y
0000 Return O if X was >= WSIZE
XEQ 67 @@@@ Y≫WSIZE?
```

```
0000 X must have been > 0 & < WSIZE
R↓
X<>Y
RCL ST Y
0000 Rotate left X bits
+/-
ROTXY
0000 Zap the right most X bits
RCL ST Z
Υ↑X
1
NOT
AND
R.TN
(SRn)
0000 DSC: Shift Y Right X Bits
0000 IN: Y: An integer
@@@@ IN: X: An integer
0000 OUT: X: IP(Y) shifted right IP(X) bit(s)
0000 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
ΙP
X<>Y
ΙP
0000 Return Y if X was negative
                @@## REQ:free42>=3.0.3
0>? ST Y
                @@## REQ:free42<3.0.3 @@## TODO: Delete when DM42 >= free42 3.0.3
XEQ 44 @@@ 0≥Y?
R.TN
X<>Y
0
@@@@ Return O if X was >= WSIZE
XEQ 67 @@@@ Y≫WSIZE?
R.TN
0000 X must have been > 0 & < WSIZE
R↓
0000 Zap the right most X bits
X<>Y
RCL ST Y
2
X<>Y
Υ↑X
1
NOT
AND
0000 Rotate right X bits
X<>Y
ROTXY
RTN
(RLn)
0000 DSC: Rotate Y Left by X bits
0000 IN: Y: An integer
0000 IN: X: An integer
@@@@ OUT: X: X rotated left IP(Y) bit(s)
0000 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
ΙP
X<>Y
0000 Return Y if X was negative
                @@## REQ:free42>=3.0.3
0≥? ST Y
XEQ 44 @@@ 0≥Y?
                @@## REQ:free42<3.0.3 @@## TODO: Delete when DM42 >= free42 3.0.3
RTN
X<>Y
0000 X was positive
```

```
+/-
ROTXY
RTN
                                                                            (RRn)
@@@@ DSC: Rotate Y Right by X bits
0000 IN: Y: An integer
@@@@ IN: X: An integer
@@@@ OUT: X: X rotated right ABS(IP(Y)) bit(s)
0000 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
ΙP
X<>Y
0000 Return Y if X was negative
              @@## REQ:free42>=3.0.3
0≽? ST Y
RTN
X<>Y
0000 X was positive
ROTXY
R.TN
(SHFXY)
@@@@ DSC: Shift X by Y Bits (left when Y negative)
0000 IN: Y: An integer
0000 IN: X: An integer
@@@@ OUT: X: Integer shifted by IP(Y) bit(s)
0000 FAQ: Uses SLn & SLr.
@@@@ LBL: Used 59
@@@@ UPD: 2021-03-20
LBL "SHFXY"
FUNC 21
                        @@## REQ:free42>=2.5.24
                        @@## REQ:free42>=3.0
L4STK
ΙP
X<0?
GTO 59
0000 X is non-negative -- shift right
XEQ "SRn"
RTN
LBL 59
0000 X negative -- shift left
+/-
XEQ "SLn"
RTN
                                                                            (RJ)
@@@@ DSC: Right Justify Bits
@@@@ IN: X: an integer
@@@@ OUT: Y: Number of shifts required to justify
0000 OUT: X: Justified number
@@@@ FAQ: Like the HP-16c's LJ funciton, but justifies in the other direction
0000 LBL: Used 56-58
@@@@ UPD: 2021-03-20
LBL "RJ"
FUNC 12
                        @@## REQ:free42>=2.5.24
L4STK
                         @@## REQ:free42>=3.0
ΙP
0
X<>Y
LBL 57
BIT?
GTO 56
GTO 58
LBL 56
       @@@@ LSB is one -- DONE!
R↓
X<>Y
RTN
     0000 LSB is zero
LBL 58
R↓
1
STO+ ST Z
ROTXY
```

```
(LJ)
0000 DSC: Left Justify Bits
0000 IN: X: an integer
@@@@ OUT: Y: Number of shifts required to justify
0000 OUT: X: Justified number
0000 FAQ: Just like the HP-16c's LJ funciton
0000 LBL: Used 64-66
@@@@ UPD: 2021-03-20
LBL "LJ"
FUNC 12
                        @@## REQ:free42>=2.5.24
L4STK
                        @@## REQ:free42>=3.0
ΤP
0
X<>Y
LBL 64
WSIZE?
1
BIT?
GTO 65
GTO 66
      @@@@ MSB is one -- DONE!
LBL 65
R↓
X<>Y
RTN
LBL 66
     0000 MSB is zero
R↓
STO+ ST Z
+/-
ROTXY
GTO 64
                                                                            (ISQRT)
0000 DSC: Integer square root
0000 IN: X: a non-negative real number
0000 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
ΙP
(ILN2)
0000 DSC: Integer base 2 log
0000 IN: X: a positive real number
@@@@ OUT: X: IP(log_2(ABS(X)))
0000 UPD: 2021-03-16
LBL "ILN2"
FUNC 11
                       @@## REQ:free42>=2.5.24
                       @@## REQ:free42>=3.0
L4STK
ABS
LN
2
LN
÷
ΙP
RTN
(MSSB)
@@@@ DSC: Position of the Most Significant Set Bit
0000 IN: X: An integer
0000 OUT: X: Position of MSSB in IP(X) or -1 if IP(X) was 0
0000 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
ΙP
```

```
-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
R.TN
(LSSB)
0000 DSC: Position of the Least Significant Set Bit
0000 IN: X: An integer
0000 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
0000 FAQ: Closely related to RJ program
LBL "LSSB"
FUNC 11
                           00## REQ:free42>=2.5.24
L4STK
                           @@## REQ:free42>=3.0
ΤP
-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
        @@@ O=Y?
                    @@## REQ:free42<3.0.3 @@## TODO: Delete when DM42 >= free42 3.0.3
                                         @@## TODO: Delete when DM42 >= free42 3.0.3
FUNC 00
                    @@## REQ: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
        @@@ O>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
                    @@## REQ:free42<3.0.3     @@## TODO: Delete when DM42 >= free42 3.0.3
X<>Y
X<0?
                    @@## REQ:free42<3.0.3 @@## TODO: Delete when DM42 >= free42 3.0.3
RTNYES
                    00## REQ:free42<3.0.3 00## TODO: Delete when DM42 >= free42 3.0.3
                    @@## 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
RTNNO
LBL 44
       @@@ 0≥Y?
FUNC OO
                    @@## REQ:free42<3.0.3 @@## TODO: Delete when DM42 >= free42 3.0.3
                    QQ## REQ: free 42 < 3.0.3 QQ## TODO: Delete when DM42 >= free 42 3.0.3
X<>Y
X<0?
                    @@## 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
@@## REQ:free42<3.0.3     @@## TODO: Delete when DM42 >= free42 3.0.3
RTNYES
RTNNO
                    @@## REQ:free42<3.0.3 @@## TODO: Delete when DM42 >= free42 3.0.3
@@@@ Menu Label: WSIZE
LBL 96
"W."
WSIZE?
AIP
BT
RTN
@@@@ Menu Label: BSIGNED
LBL 97
FS? 78
"S·+/-"
FC? 78
"S:+"
R.TN
@@@@ Menu Label: BWRAP
LBL 98
FS? 79
"BWR•"
FC? 79
"BWRP"
```

```
@@@@ Menu Label: BVIEW
LBL 99
FC? 02
"BVA•"
FS? 02
"BVA"
RTN
0000 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
0000 Toggle 02
LBL 74
FS?C 02
RTN
SF 02
RTN
@@@@ Menu Action BINM
LBL 75
EXTTALL.
"Press R/S To"
"⊢ Return"
AVIEW
HEXM
```

```
STOP
RTN
@@@@ DSC: Test if Y<WSIZE
@@@@ NAM: Y>WSIZE? 67
0000 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)
0000 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
           0000 X X Y
XEQ "MSKR"
            0000 M X Y
           0000 X M Y
X<>Y
1
LSTO "_MBC"
1000
LSTO "_CTR"
R↓
           @@@@ M Y
RCL ST Y
            @@@@ Y M Y
            @@@@ N Y
OR
X<>Y
            @@@@ OLD NEW
LBL 51
           @@@@ CTR OLD NEW
RCL "_CTR"
ΙP
BIT?
GTO 52
R↓
X<>Y
RCL "_CTR"
ΙP
RCL- "_MBC"
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
0000 IN: X: An integer
0000 OUT: X: An integer
@@@@ UPD: 2021-04-21
LBL "REVB"
FUNC 11
L4STK
```

WSIZE? XEQ "REVNB"

```
(POPRB)
@@@@ DSC: Shift rihtmost X bits off Y.
0000 IN: Y: An integer
0000
                   X: An integer (number of bits)
0000 OUT: Y: Part of Y left
0000
             X: Part of Y shifted off
LBL "POPRB"
FUNC 22
L4STK
                          0000 Y X Y ?
RCL ST Y
                        0000 X Y X Y
RCL ST Y
                        0000 LFT X Y Y
XEQ "SRn"
                      0000 Y LFT X Y
RCL ST Z
RCL ST Z
XEQ "MSKR" @@@@ MSK Y LFT X
AND
                          0000 POP LFT X Y
R.TN
occorrections of the contraction of the contracti
                                                                                                                                                                                                    (POPLB)
0000 DSC: Shift leftmost X bits off Y.
0000 IN: Y: An integer
0000 X: An integer (number of bits)
0000 OUT: Y: Part of Y left
0000 X: Part of Y shifted off
LBL "POPLB"
FUNC 22
L4STK
RCL ST Y
                           0000 Y X Y ?
                         0000 X Y X Y
RCL ST Y
                       0000 LFT X Y Y
XEQ "SLn"
                      0000 Y LFT X Y
RCL ST Z
RCL ST Z 0000 X Y LFT X XEQ "MSKL" 0000 MSK Y LFT X
                          0000 PS LFT X Y
AND
RCL ST Z
+/-
ROTXY
                          @@@@ POP LFT X Y
R.TN
(BVIEW)
0000 DSC: Print binary numbers (up to 64-bit)
0000 IN: X: An integer
@@@@ OUT: Nothing -- prints to screen
@@@@ FLG: 81: TEMP: Set: We are on DM42;
                                                                                                         Clear: We are not on DM42
0000 FLG: 01: PREF: Reserved
0000 FLG: 02: PREF: Set: BVIEW prints WSIZE digits; Clear: BVIEW prints 64 digits
0000 FAQ: If XEQ while system BASE menu is active, returns to same menu.
0000 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
Ω
SF 25
BIT?
FS2C 25
GTO 79
R.↓
"ERR: Bad Int!"
AVIEW
RTN
LBL 79
RT
LSTO "_X"
20
1
FS? 68
STO+ ST Y
FS? 70
STO+ ST Y
FS? 71
STO+ ST Y
                               @@@@ 20:DECN 21:BINM 22:OCTM 23:HEXM
R.J
LSTO "_SBMD"
```

```
R↓
CF 81
SF 25
RCL "GrMod"
FS?C 25
SF 81
FC? 81
GTO 88
0000 We are on a DM42
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?
0000 Figure out X&Y coordinates for digit
RCL "_CTR"
                   0000 IF
ΙP
32
X>Y?
GTO 73
GTO 83
LBL 73
                    @@@@ IF-THEN FIRST ROW
RCL "_CTR"
4
1
GTO 90
LBL 83
                    @@@@ IF-ELSE SECOND ROW
9
RCL "_CTR"
32
4
×
1
LBL 90
                   0000 IF-END
0000 Figure out current bit value
63
RCL "_CTR"
                     @@@@ IF-BEGIN
ΙP
WSIZE?
X>Y?
GTO 80
GTO 81
LBL 80
                    0000 IF-THEN
R↓
+/-
                    @@@@ IF-BEGIN
1
Х<>У
ROTXY
RCL "_X"
AND
X=0?
GTO 85
GTO 86
LBL 85
                    @@@@ IF-THEN bit is 0
R↓
0
GTO 87
LBL 86
                    0000 IF-ELSE bit is 1
R↓
1
LBL 87
LSTO "_CB"
GTO 82
```

```
LBL 81
                   0000 IF-ELSE
R↓
R↓
0000 Bit beyond WSIZE padding character
0
LSTO "_CB"
FS? 02
GTO 84
LBL 82
                   0000 IF-END
R↓
0000 Figure out grouping
RCL "_CTR"
                    @@@@ IF-START grouping
ΙP
4
ΙP
2
FP
X=0?
GTO 55
GTO 68
LBL 55
                   @@@@ IF-THEN
R.J
GTO 77
LBL 68
                   0000 IF-ELSE
R↓
2
LBL 77
                   0000 IF-END
0000 Compute digit character code
RCL+ "_CB"
0000 Draw it
XEQ 78
LBL 84
ISG "_CTR"
GTO 91
FC? 81
GTO 89
0000 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↓
0000 Recall original X
RCL "_X"
RTN
0000 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
0000
         X: Character number
@@@@ OUT: No return
@@@@ UPD: 2021-04-11
0000 BUG: Characters can be *VERY* tiny in high resolution modes on DM42
0000 Characters can be underlined
0000 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:
0000
       00 01
0000
         0 1
\ensuremath{\text{@@@@}} Add 2 to the above character number for the underlined version
            @@## REQ:free42>=2.5.24
FUNC 30
L4STK
            @@## REQ:free42>=3.0
ΙP
60
XEQ IND ST X
R↓
AGRAPH
RTN
LBL 60
           0000 CHAR: 0
"•μ•"
           @@@@ #b11111 #b10001 #b11111
RTN
           0000 CHAR: 1
LBL 61
"£•←"
            @@@@ #b10010 #b11111 #b10000
RTN
LBL 62
            @@@@ CHAR: O
"_Q_"
            @@@@ #b1011111 #b1010001 #b1011111
RTN
LBL 63
            0000 CHAR: 1
"R_P"
            @@@@ #b1010010 #b1011111 #b1010000
RTN
```

#### 

# 4 EOF