

SNUMREP

Rx32: number to IEEE 32 (0.8s)	9: 8: 7: 6: 5: 4: 3: 2: 1: - .015625	1.01111001.000000000000 000000000000
ViewN: view number (0.1s)	1: "1.01111001.00000000..." Exnum ViewN Rx32 Rx64 R+HEX rxb	
Rx32: large integer to IEEE 64 (0.6s) and back (2.5s)	9: 8: 7: 6: 5: 4: 3: 2: 1: -29343216566272	1.10101011.10101011000 000000000000
ViewN: view number (0.1s)	2: "1.10101011.101010..." 1: "-2.93432165662E13" Exnum ViewN Rx32 Rx64 R+HEX rxb	
Rx64: number to IEEE 64 (3s) and back (4.5s)	9: 8: 7: 6: 5: 4: 3: 2: 1: 1.57775671636E-4	0.01111110010.01001010 1110000100101111000101 1101100000001111010000
ViewN: view number (0.1s)	2: "0.01111110010.010..." 1: ".0001577757" Exnum ViewN Rx32 Rx64 R+HEX rxb	
RxHEX: IEEE to hex number (0.1s)	9: 8: 7: 6: 5: 4: 3: 2: 1: "1.10101011.101010..." "# D5D58000h"	9: 8: 7: 6: 5: 4: 3: 2: 1: .333333333333 "# 0.0101010101010..."
rxb: unsigned real to base (2.8s)	Exnum ViewN Rx32 Rx64 R+HEX rxb	Exnum ViewN Rx32 Rx64 R+HEX rxb
Stws: set binary word size to 8	set binary word size (1-64) actual size: 64	9: 8: 7: 6: 5: 4: 3: 2: 1: 46 "# 00101110b"
NxUnsign: positive integer to unsigned number (0.5s)	8 Stws NxUnsign NxSign NxOne NxTwo NxExcess	Stws NxUnsign NxSign NxOne NxTwo NxExcess
NxSign: integer to sign + magnitude (0.5s)	signed number	ones' complement
NxOnecmp: integer to ones' complement (0.2s)	7: 6: 5: 4: 3: 2: 1: -46 "# 10101110b"	7: 6: 5: 4: 3: 2: 1: -46 "# 11010001b"
NxTwocmp: integer to two's complement (0.2s)	Stws NxUnsign NxSign NxOne NxTwo NxExcess	Stws NxUnsign NxSign NxOne NxTwo NxExcess
NxExcess: integer to excess (0.2s)	two's complement	excess
	7: 6: 5: 4: 3: 2: 1: -46 "# 11010010b"	7: 6: 5: 4: 3: 2: 1: -46 "# 01010001b"
	Stws NxUnsign NxSign NxOne NxTwo NxExcess	Stws NxUnsign NxSign NxOne NxTwo NxExcess
HelpSNUMREP: help	SNUMREP SIGNED NUMBER REPR x=real, n=integer, #=base number Exnum - + examples choosebox ViewN x + view large number, [ENTER] quits Rx32 x ↔ IEEE 32 bits sign.exp.frac=1.8.23 Rx64 x ↔ IEEE 64 bits sign.exp.frac=1.11.52 R+HEX IEEE → hex number str. rxb x ↔ # with fractions GRAPH	rxb x ↔ # with fractions Stws - + set bin. word size For Nx... programs NxUnsign n ↔ unsigned number NxSign n ↔ sign + number NxOnecmp n ↔ ones' complement NxTwoCmp n ↔ two's complement NxExcess n ↔ excess Bconv x + #, subprog of rxb information: (1=sign -, 0=sign +) GRAPH