Boolean
AND
OR
NAND
NOR
NOT
MOD
XOR
shl
shr
asl
asr
rol
ror
y<<x
y>>x

Base/Data type
bsCvt
asInt
asFlt
sBit

Basic
+
-
X
/
±
1/x
x!

Trigonometry
sin (asin)
cos (acos)
tan (atan)
csc (acsc)
sec (asec)
cot (acot)
sinh (asinh)
cosh (acosh)
tanh (atanh)
csch (acsch)
sech (asech)
coth (acoth)

Conversion
bsCvt
→.hms (.hms→)
.hms+
.hms-
→h.ms (h.ms→)
h.ms+
h.ms-
→rθ (→xy)
→rθφ (→xyz)
→ρφz (→xyz)
→rθφ (→ρφz)

Power
ln (e^x)
log2 (2^x)
log10 (10^x)
x^2 (√)
x^3 (∛)
√ (x^2)
y^x (y^1/x)
sinh (asinh)
cosh (acosh)
tanh (atanh)
csch (acsch)
sech (asech)
coth (acoth)

Register
y→[x] (s[x]→)
[x]→x (→s[x])
count
x→% (%→X)

Constants
π
e
h
c
∞ (-∞)
NAN (-NAN)

Stats
mean (mdian)
stdev (psdev)
∑S[] (∏S[])
∑S^2
count
x!
yChsX

Financial
X * %
x→% (%→X)
%
Δ%
fVal
pVal

AND

Function Performed:
R = X & Y

X	Y	R
0	0	0
0	1	0
1	0	0
1	1	1

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer
Y must be integer
Word Size Extension Applies
Signed from X

OR

Function Performed:
R = X | Y

X	Y	R
0	0	0
0	1	1
1	0	1
1	1	1

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer
Y must be integer
Word Size Extension Applies
Signed from X

NAND

Function Performed:
R = X & Y

X	Y	R
0	0	1
0	1	1
1	0	1
1	1	0

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer
Y must be integer
Word Size Extension Applies
Signed from X

NOR

Function Performed:
R = X & Y

X	Y	R
0	0	1
0	1	0
1	0	0
1	1	0

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer

Y must be integer

Word Size Extension Applies

Signed from X

NOT

Function Performed:
R = ~X

X	R
0	1
1	0

Consumed:
X

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer

Signed from X

MOD

Function Performed:
R = X % Y

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer

Y must be integer

Result will be (+)

XOR

Function Performed:
R = X & Y

X	Y	R
0	0	1
0	1	0
1	0	0
1	1	0

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X must be integer

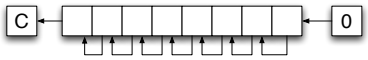
Y must be integer

Word Size Extension Applies

Signed from X

shl

Function Performed:
R = X << 1



Consumed:
X, Y

Results in:
X

Alt Function:
NO

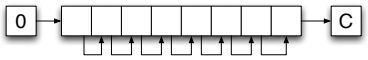
Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

srh

Function Performed:
R = X >> 1



Consumed:
X

Results in:
X

Alt Function:
NO

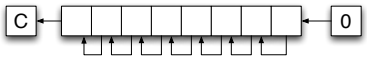
Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

asl

Function Performed:
R = X << 1



Consumed:
X, Y

Results in:
X

Alt Function:
NO

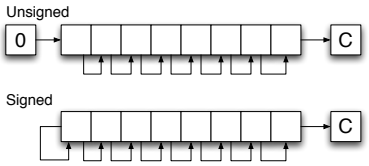
Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

asr

Function Performed:
R = X >> 1



Consumed:
X, Y

Results in:
X

Alt Function:
NO

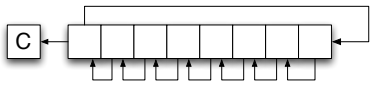
Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

rol

Function Performed:
R = (X << 1) | X_{MSB} into X_{LSB}



Consumed:
X, Y

Results in:
X

Alt Function:
NO

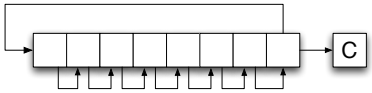
Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

ror

Function Performed:
R = (X >> 1) | X_{LSB} into X_{MSB}



Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

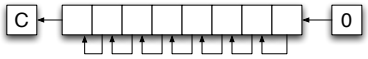
O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

y<<x

Function Performed:
R = X << Y

Operation Y times:



Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

Y must be integer

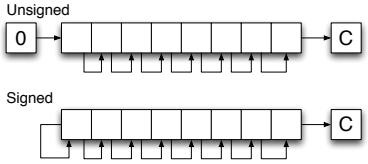
Word size from X

Signed from X

y>>x

Function Performed:
R = X >> Y

Operation Y times:



Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	X

Restrictions:
X must be integer

Y must be integer

Word size from X

Signed from X

bsCvt

Function Performed:
Convert Y to Base X

Consumed:
X, Y

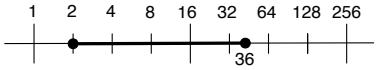
Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:
X Must be Integer and



asFlt

Function Performed:
Assign the value in X to a floating point without any conversion.

Consumed:
X

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
			E	

Restrictions:
No range checking is done. Result my be NAN, ∞ or other non-sensical value.

Results are best when the input is a 64 bit unsigned number.

asInt

Function Performed:
Assign a floating point value in X to a 64 unsigned Integer without any conversion.

Consumed:
X

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C

Restrictions:
This will show the internal representation of a floating point number in 64 bit hex format.

sBit

Function Performed:
Set the significant bits used on the mantissa output of floating point numbers. The default value is 48 bits.

A value of 0 will reset the significant bits back to the default of 48.

	Radix			
bits	2	8	10	16
4	4	2	2	1
8	8	3	3	2
12	12	4	4	3
16	16	6	5	4
20	20	7	7	5
24	24	8	8	6
32	32	11	10	8
40	40	14	13	10
48	48	16	15	12

Consumed:
X

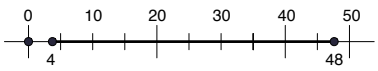
Results in:
None

Alt Function:
NO

Flags:

O	U	E	N	C
		X		

Restrictions:
X is integer and:



+

Function Performed:
R = X + Y

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
X	X		X	

Restrictions:
When both X.Y are Integer, result is Integer. See NOTE.

When either X.Y is Floating point, result is Floating Point

NOTE: When Auto Promotion is selected in the defaults screen, the YASC will treat all integers as floating point.

-

Function Performed:
R = X - Y

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
X	X		X	

Restrictions:
When both X.Y are Integer, result is Integer. See NOTE.

When either X.Y is Floating point, result is Floating Point

NOTE: When Auto Promotion is selected in the defaults screen, the YASC will treat all integers as floating point.

X

Function Performed:
R = X * Y

Consumed:
X, Y

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C
X	X		X	

Restrictions:
When both X.Y are Integer, result is Integer. See NOTE.

When either X.Y is Floating point, result is Floating Point

NOTE: When Auto Promotion is selected in the defaults screen, the YASC will treat all integers as floating point.

/

Function Performed:

$R = \frac{Y}{X}$

Consumed:

X, Y

Results in:

X

Alt Function:

NO

Flags:

O	U	E	N	C
X	X	X	X	

Restrictions:

When both X.Y are Integer, result is Integer. See NOTE.

When either X.Y is Floating point, result is Floating Point

NOTE: When Auto Promotion is selected in the defaults screen, the YASC will treat all integers as floating point.

±

Function Performed:

Change Sign

Consumed:

X

Results in:

X

Alt Function:

NO

Flags:

O	U	E	N	C
			X	

Restrictions:

Does not end the edit cycle

While in Edit Mode:

Will perform a CHS on the exponent if entering an exponent.

Will perform a CHS on the mantissa when no exponent exists.

When not in Edit Mode:

Will perform a CHS on the mantissa when no exponent exists.

When X is signed Integer:

Will perform the 2's complement and show the "-" sign.

"1231" will become "-1231"

When X is unsigned Integer:

Will perform 2's compliment.

"1" will become 65535 if word size is set to 16.

1/x

Function Performed:

$R = \frac{1}{X}$

Consumed:

X

Results in:

X

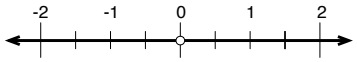
Alt Function:

NO

Flags:

O	U	E	N	C
		X	X	

Restrictions:



x!

Function Performed:

$$R = \prod_{n=1}^{x} n$$

R = 1 when x = 0

Consumed:
X

Results in:
X

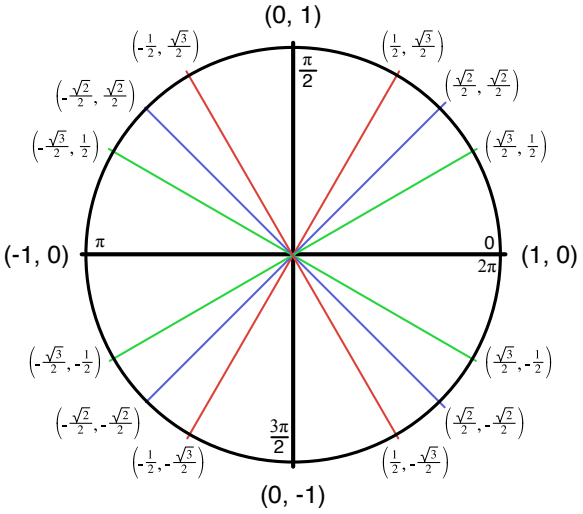
Alt Function:
NO

Flags:

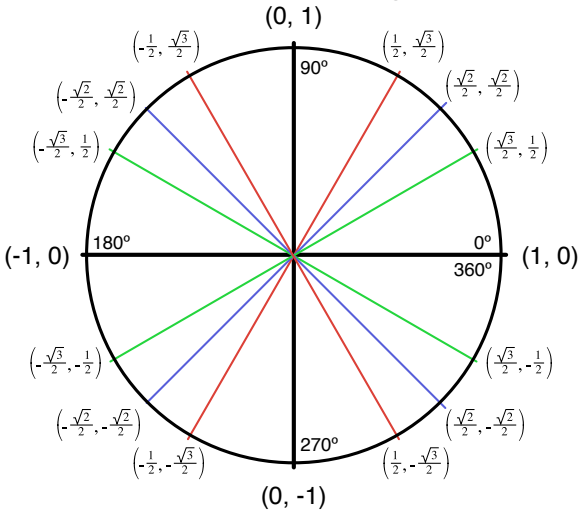
O	U	E	N	C
X		X		

Restrictions:
X is a positive Integer.

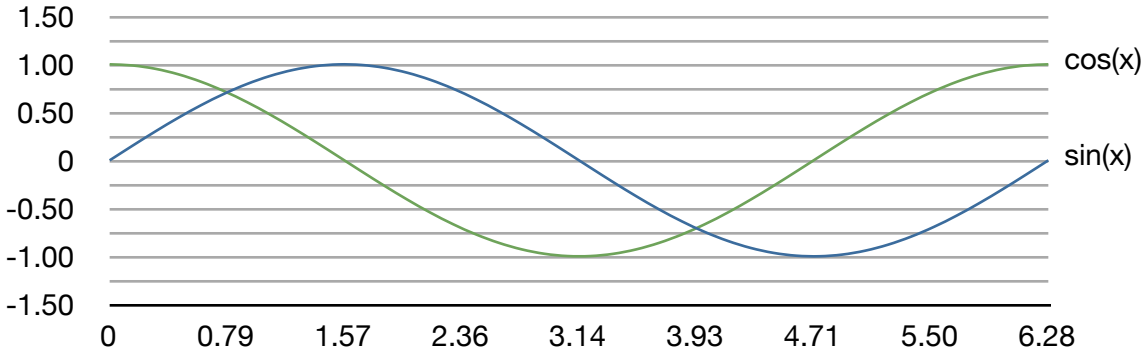
Unit Circle for Radians



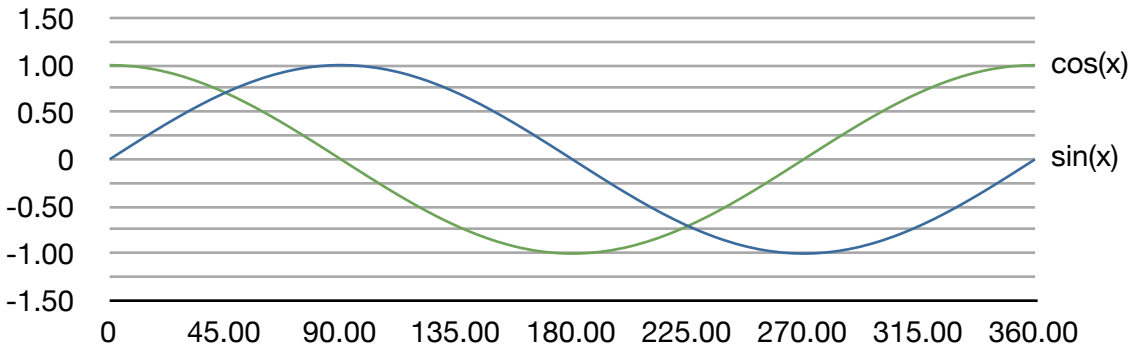
Unit Circle for Degrees



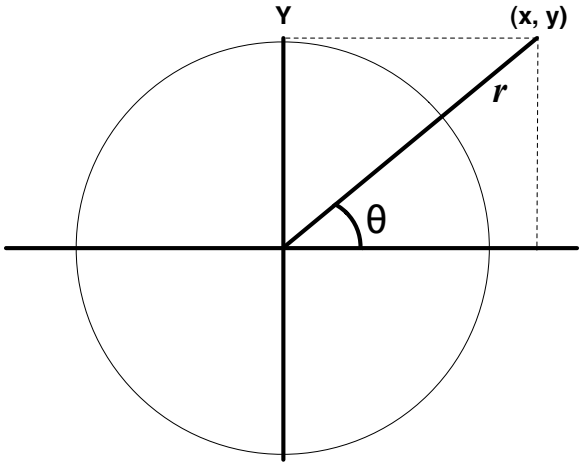
Sin and Cos for Selected Radians



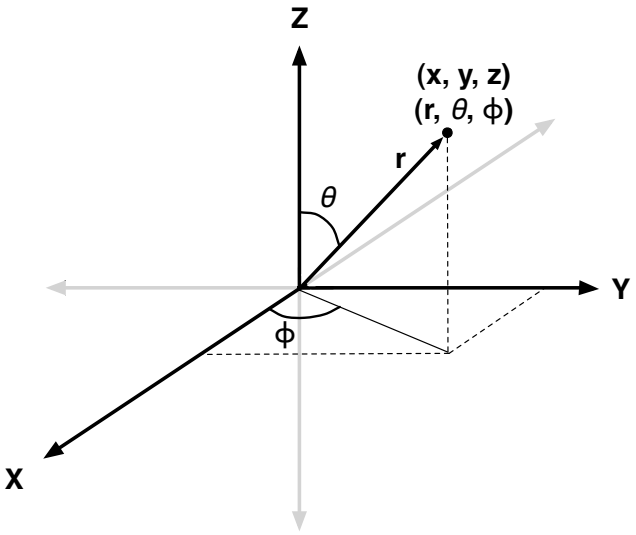
Sin and Cos for Selected Degrees



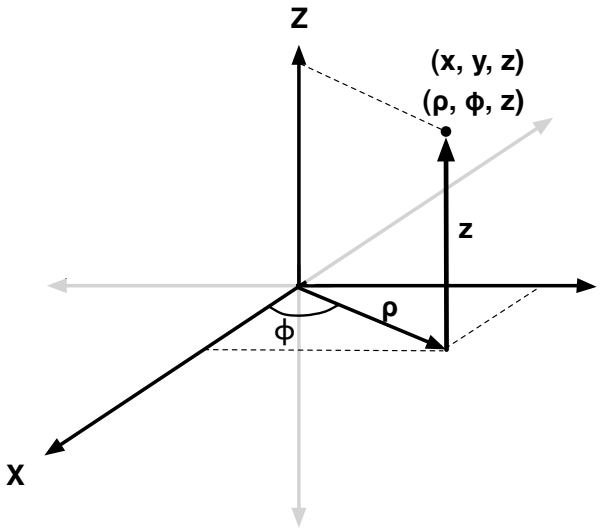
Polar/Cartesian Relationship



Sperical/Cartesian Relationship



Cylindrical/Cartesian Relationship



sin (asin)

Function Performed:

$R = sin(X)$

$R' = asin(X)$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
sin(X)

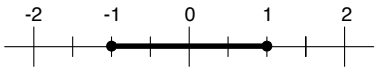
O	U	E	N	C
			X	

asin(X)

O	U	E	N	C
		X	X	

Restrictions:
sin - the domain of real numbers.

asin:



NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

cos (acos)

Function Performed:

$R = cos(X)$

$R' = acos(X)$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
cos(X)

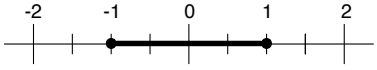
O	U	E	N	C
			X	

acos(X)

O	U	E	N	C
		X	X	

Restrictions:
cos - the domain of real numbers.

acos:



NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

tan (atan)

Function Performed:

$R = \frac{sin(x)}{cos(x)}$

$R' = acos(X)$

Consumed:
X

Results in:
X

Alt Function:
YES

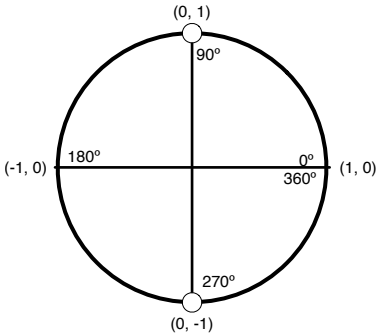
Flags:
tan(X)

O	U	E	N	C
		X	X	

atan(X)

O	U	E	N	C
			X	

Restrictions:
tan:



atan - the domain of real numbers.

NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

csc (acsc)

Function Performed:

$$R = \frac{1}{sin(x)}$$

$$R' = 2 \cdot atan\left(\frac{\frac{1}{x}}{1 + \sqrt{1 - \left(\frac{1}{x^2}\right)}}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

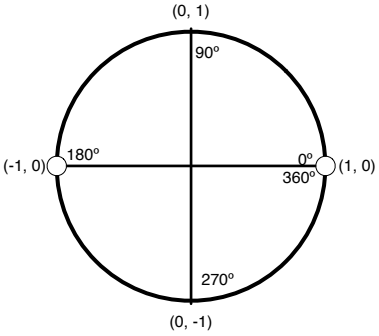
Flags:
csc(X)

O	U	E	N	C
		X	X	

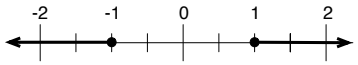
acsc(X)

O	U	E	N	C
		X	X	

Restrictions:
csc:



acsc:



NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

sec (asec)

Function Performed:

$$R = \frac{1}{cos(x)}$$

$$R' = \begin{cases} \pi & -1 = x \\ 2 \cdot atan\left(\frac{\sqrt{1 - \left(\frac{1}{x}\right)}}{1 + \left(\frac{1}{x}\right)}\right) & -1 < x \le 1 \end{cases}$$

Consumed:
X

Results in:
X

Alt Function:
YES

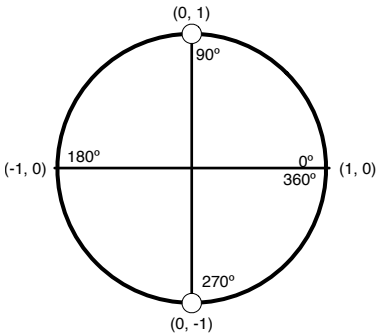
Flags:
sec(X)

O	U	E	N	C
		X	X	

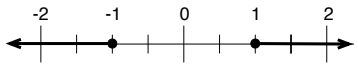
asec(X)

O	U	E	N	C
		X	X	

Restrictions:
sec:



asec:



NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

cot (acot)

Function Performed:

$$R = \frac{cos(x)}{sin(x)}$$

$$R' = \begin{cases} \pi + atan\left(\frac{1}{x}\right) & x < 0 \\ atan\left(\frac{1}{x}\right) & x > 0 \end{cases}$$

Consumed:
X

Results in:
X

Alt Function:
YES

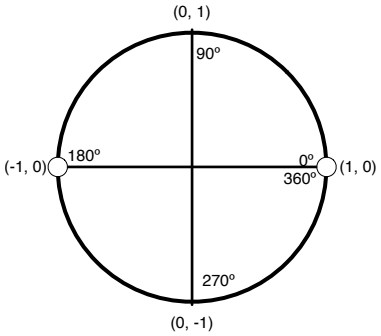
Flags:
cot(X)

O	U	E	N	C
		X	X	

acot(X)

O	U	E	N	C
			X	

Restrictions:
cot:



acot - the domain of real numbers.

NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

→.hms (.hms→)

Function Performed:

→.hms

Convert decimal hours to:

ddd.hhh:mmm:sss.fracs

Where:

ddd = Days/Degrees

hhh = Hours (0-24)

mmm = Minutes (0-60)

sss = Seconds (0-60)

fracs = fractions of seconds

(.hms→)

Convert from:

ddd.hhh:mmm:sss.fracs

to decimal hours.

Consumed:

X

Results in:

X

Alt Function:

YES

Flags:

→.hms

.hms→

O	U	E	N	C
			X	

Restrictions:

None

NOTE: An input of (hhh = 100) is allowed. Same is true for mmm and sss.

NOTE: The output is formatted specifically for Radix 10 math.

.hms+

Function Performed:

Add .hms in X to .hms in Y.

ddd.hhh:mmm:sss.fracs

Where:

ddd = Days/Degrees

hhh = Hours (0-24)

mmm = Minutes (0-60)

sss = Seconds (0-60)

fracs = fractions of seconds

Consumed:

X, Y

Results in:

X

Alt Function:

NO

Flags:

.hms+

O	U	E	N	C
			X	

Restrictions:

None

NOTE: An input of (hhh = 100) is allowed. Same is true for mmm and sss.

NOTE: The output is formatted specifically for Radix 10 math.

NOTE: Result will normalize hhh, mmm and sss to be within 0-24, 0-60 and 0-60 respectively.

.hms-

Function Performed:

Subtract .hms in X from .hms in Y.

ddd.hhh:mmm:sss.fracs

Where:

ddd = Days/Degrees

hhh = Hours (0-24)

mmm = Minutes (0-60)

sss = Seconds (0-60)

fracs = fractions of seconds

Consumed:

X, Y

Results in:

X

Alt Function:

NO

Flags:

.hms-

O	U	E	N	C
			X	

Restrictions:

None

NOTE: An input of (hhh = 100) is allowed. Same is true for mmm and sss.

NOTE: The output is formatted specifically for Radix 10 math.

NOTE: Result will normalize hhh, mmm and sss to be within 0-24, 0-60 and 0-60 respectively.

→h.ms (h.ms→)

Function Performed:

→h.ms

Convert decimal hours to:

hhh.mmm:sss.fracs

Where:

hhh = Hours

mmm = Minutes (0-60)

sss = Seconds (0-60)

fracs = fractions of seconds

(h.ms→)

Convert from:

ddd.hhh:mmm:sss.fracs

to decimal hours.

Consumed:

X

Results in:

X

Alt Function:

YES

Flags:

→h.ms

h.ms→

O	U	E	N	C
			X	

Restrictions:

None

NOTE: An input of (hhh = 100) is allowed. Same is true for mmm and sss.

NOTE: The output is formatted specifically for Radix 10 math.

h.ms+

Function Performed:

Add h.ms in X to h.ms in Y.

hhh.mmm:sss.fracs

Where:

hhh = Hours

mmm = Minutes (0-60)

sss = Seconds (0-60)

fracs = fractions of seconds

Consumed:

X, Y

Results in:

X

Alt Function:

NO

Flags:

hms+

O	U	E	N	C
			X	

Restrictions:

None

NOTE: An input of (hhh = 100) is allowed. Same is true for mmm and sss.

NOTE: The output is formatted specifically for Radix 10 math.

NOTE: Result will normalize mmm and sss to be within 0-60 and 0-60 respectively.

h.ms-

Function Performed:

Subtract h.ms in X from h.ms in Y.

hhh.mmm:sss.fracs

Where:

hhh = Hours

mmm = Minutes (0-60)

sss = Seconds (0-60)

fracs = fractions of seconds

Consumed:

X, Y

Results in:

X

Alt Function:

NO

Flags:

hms+

O	U	E	N	C
			X	

Restrictions:

None

NOTE: An input of (hhh = 100) is allowed. Same is true for mmm and sss.

NOTE: The output is formatted specifically for Radix 10 math.

NOTE: Result will normalize mmm and sss to be within 0-60 and 0-60 respectively.

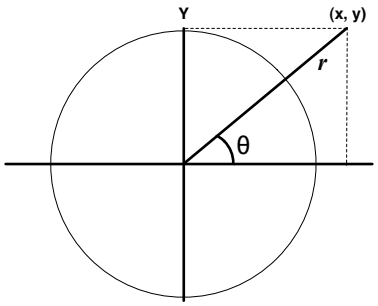
→rθ (→xy)

Function Performed:

$$R = \begin{cases} r_x = \sqrt{x^2 + y^2} \\ \theta_y = \begin{cases} 0 & x = y = 0 \\ atan2\left(\frac{y}{x}\right) & x \neq 0 \text{ and } y \neq 0 \end{cases} \end{cases} \quad -\pi \le \theta \le \pi$$

$$R' = \begin{cases} x_r = r \cdot cos(\theta) \\ y_\theta = r \cdot sin(\theta) \end{cases}$$

Polar/Cartesian Relationship



Consumed:
X, Y

Results in:
X, Y

Alt Function:
YES

Flags:
→rθ

→xy

O	U	E	N	C
			X	

Restrictions:

None

NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

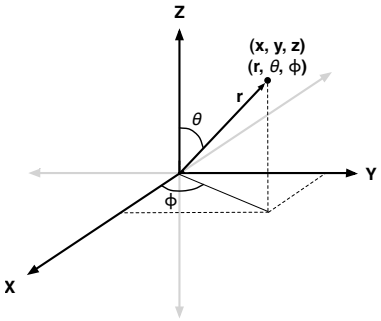
→rθφ (→xyz)

Function Performed:

$$R = \begin{cases} r_x = \sqrt{x^2 + y^2 + z^2} \\ \theta_y = acos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \\ \varphi_z = atan2\left(\frac{y}{x}\right) \end{cases}$$

$$R' = \begin{cases} x_r = r \cdot sin(\theta)cos(\varphi) \\ y_\theta = r \cdot sin(\theta)sin(\varphi) \\ z_\varphi = r \cdot cos(\theta) \end{cases}$$

Sperical/Cartesian Relationship



Consumed:
X, Y, Z

Results in:
X, Y, Z

Alt Function:
YES

Flags:
→rθφ

→xyz

O	U	E	N	C
			X	

Restrictions:

None

NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

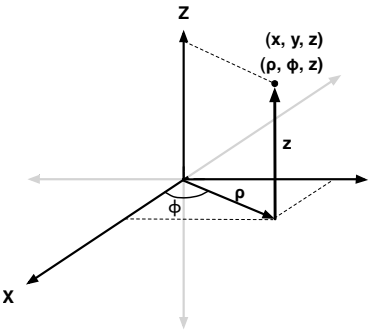
→ρφz (→xyz)

Function Performed:

$$R = \begin{cases} \varrho_x = \sqrt{x^2 + y^2} \\ \varphi_y = \begin{cases} 0 & x = y = 0 \\ \text{asin}\left(\frac{y}{\sqrt{x^2 + y^2}}\right) & x \ge 0 \\ \text{-asin}\left(\frac{y}{\sqrt{x^2 + y^2}}\right) + \pi & x < 0 \end{cases} \\ z_z = z \end{cases}$$

$$R' = \begin{cases} x_\varrho = \varrho \cdot cos(\varphi) \\ y_\varphi = \varrho \cdot sin(\varphi) \\ z_z = z \end{cases}$$

Cylindrical/Cartesian Relationship



Consumed:
X, Y

Results in:
X, Y, Z

Alt Function:
YES

Flags:
→ρφz

→xyz

O	U	E	N	C
			X	

Restrictions:

None

NOTE: YASC uses the RAD/ DEG button to determine any scaling of the answer/input.

→rθφ (→ρφz)

Function Performed:

$$R = \begin{cases} r_{\varrho(x)} = \sqrt{\varrho_x^2 + z^2} \\ \theta_{\varphi(y)} = asin\left(\frac{z}{\sqrt{\varrho_x^2 + z^2}}\right) \\ \varphi_{\varphi(z)} = \varphi_y \end{cases}$$

$$R' = \begin{cases} \varrho_{r(x)} = r_x \cdot cos(\theta_y) \\ \varphi_{\theta(y)} = \varphi_z \\ z_{\varphi(z)} = r_x \cdot sin(\theta_y) \end{cases}$$

NOTE: Ref (→rθφ) and (→ρφz) for coordinate configuration.

Consumed:
X, Y, Z

Results in:
X, Y, Z

Alt Function:
YES

Flags:
→ρφz

→xyz

O	U	E	N	C
			X	

Restrictions:
None

NOTE: YASC uses the RAD/DEG button to determine any scaling of the answer/input.

sinh (asinh)

Function Performed:

$$R = \tfrac{1}{2}(e^X - e^{-X})$$

$$R' = ln\left(x + \sqrt{x^2 + 1}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
sinh(X)

O	U	E	N	C
			X	

asinh(X)

O	U	E	N	C
			X	

Restrictions:
sinh - the domain of real numbers.

asinh - the domain of real numbers.

cosh (acosh)

Function Performed:

$$R = \tfrac{1}{2}(e^X + e^{-X})$$

$$R' = ln\left(x + \sqrt{x^2 - 1}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
cosh(X)

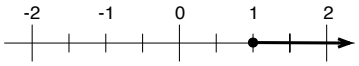
O	U	E	N	C
			X	

acosh(X)

O	U	E	N	C
		X	X	

Restrictions:
cosh - the domain of real numbers.

acosh:



tanh (atanh)

Function Performed:

$$R = \frac{sinh(x)}{cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

$$R' = \frac{1}{2} ln\left(\frac{1 + x}{1 - x}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
tanh(X)

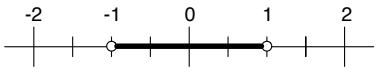
O	U	E	N	C
			X	

atanh(X)

O	U	E	N	C
		X	X	

Restrictions:
tanh - the domain of real numbers.

atanh:



csch (acsch)

Function Performed:

$$R = \frac{2}{(e^{X} - e^{-X})}$$

$$R' = ln\left(\frac{1}{X} + \frac{\sqrt{1 + X^2}}{|X|}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

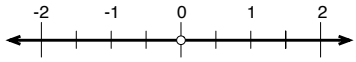
Flags:
csch(X)

O	U	E	N	C
		X	X	

acsch(X)

O	U	E	N	C
			X	

Restrictions:
csch:



acsch - the domain of real numbers.

sech (asech)

Function Performed:

$$R = \frac{2}{(e^{X} + e^{-X})}$$

$$R' = ln\left(\frac{1 + \sqrt{1 - x^2}}{x}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
sech(X)

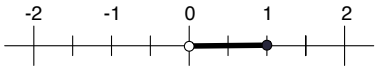
O	U	E	N	C
			X	

asech(X)

O	U	E	N	C
		E	X	

Restrictions:
sech - the domain of real numbers.

asech:



coth (acoth)

Function Performed:

$$R = \frac{cosh(x)}{sinh(x)} = \frac{e^{2x} + 1}{e^{2x} - 1}$$

$$R' = \frac{1}{2} ln\left(\frac{X+1}{X-1}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

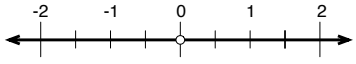
Flags:
coth(X)

O	U	E	N	C
		X	X	

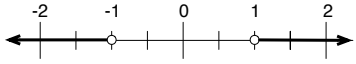
acoth(X)

O	U	E	N	C
		X	X	

Restrictions:
coth



acoth:



x^2 (√)

Function Performed:

$$R = X^2$$

$$R' = \sqrt{X}$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
x^2

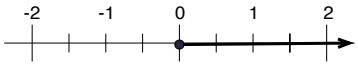
O	U	E	N	C
X	X			

√

O	U	E	N	C
		X		

Restrictions:
x^2 - The domain of real numbers.

√ **:**



√ (x^2)

Function Performed:

$$R = \sqrt{X}$$

$$R' = x^2$$

Consumed:
X

Results in:
X

Alt Function:
YES

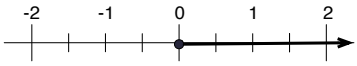
Flags:
√

O	U	E	N	C
		X		

x^2

O	U	E	N	C
X	X			

Restrictions:
√ **:**



x^2 - The domain of real numbers.

x^3 (∛)

Function Performed:

$$R = x^3$$

$$R' = \sqrt[3]{x}$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
x^3

O	U	E	N	C
X	X		X	

∛

O	U	E	N	C
			X	

Restrictions:
x^3 - The domain of real numbers.

∛ - The domain of real numbers.

y^x (y^1/x)

Function Performed:

$$R = Y^X$$

$$R' = \sqrt[x]{Y}$$

Consumed:
X, Y

Results in:
X

Alt Function:
YES

Flags:
y^x

O	U	E	N	C
X	X	X	X	

y^1/x

O	U	E	N	C
X	X	X	X	

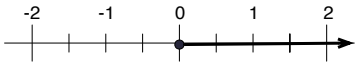
Restrictions:
y^x

If X = 0 R = 1

If Y = 0 R = 0

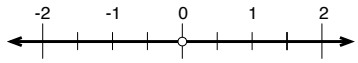
X - The domain of real numbers.

Y:

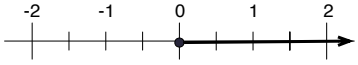


y^1/x

X:



Y:



If X = 0 R = 1

If Y = 0 R = 0

ln (e^x)

Function Performed:

$$R = ln(X)$$

$$R' = e^X$$

Consumed:
X

Results in:
X

Alt Function:
YES

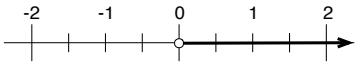
Flags:
ln

O	U	E	N	C
		X	X	

e^x

O	U	E	N	C
X	X			

Restrictions:
ln:



e^x - The domain of real numbers.

log2 (2^x)

Function Performed:

$$R = log_2(X)$$

$$R' = 2^X$$

Consumed:
X

Results in:
X

Alt Function:
YES

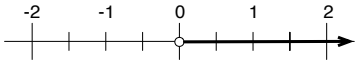
Flags:
log2

O	U	E	N	C
		X	X	

2^x

O	U	E	N	C
X	X			

Restrictions:
log2:



2^x - The domain of real numbers.

log10 (2^x)

Function Performed:

$$R = log_{10}(X)$$

$$R' = 10^{X}$$

Consumed:
X

Results in:
X

Alt Function:
YES

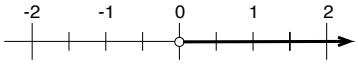
Flags:
log10

O	U	E	N	C
		X	X	

10^x

O	U	E	N	C
X	X			

Restrictions:
log10:



10^x - The domain of real numbers.

sinh (asinh)

Function Performed:

$$R = \tfrac{1}{2}(e^{X} - e^{-X})$$

$$R' = ln\left(x + \sqrt{x^2 + 1}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
sinh

O	U	E	N	C
			X	

asinh

O	U	E	N	C
			X	

Restrictions:
sinh - the domain of real numbers.

asinh - the domain of real numbers.

cosh (acosh)

Function Performed:

$$R = \tfrac{1}{2}(e^{X} + e^{-X})$$

$$R' = ln\left(x + \sqrt{x^2 - 1}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
cosh

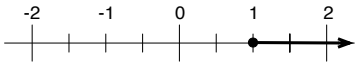
O	U	E	N	C
			X	

acosh

O	U	E	N	C
		X	X	

Restrictions:
cosh - the domain of real numbers.

acosh:



tanh (atanh)

Function Performed:

$$R = \frac{sinh(x)}{cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

$$R' = \frac{1}{2} ln\left(\frac{1 + x}{1 - x}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
tanh

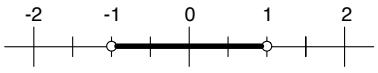
O	U	E	N	C
			X	

atanh

O	U	E	N	C
		X	X	

Restrictions:
tanh - the domain of real numbers.

atanh:



csch (acsch)

Function Performed:

$$R = \frac{2}{(e^{X} - e^{-X})}$$

$$R' = ln\left(\frac{1}{X} + \frac{\sqrt{1 + X^2}}{|X|}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

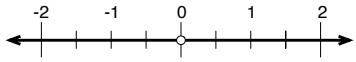
Flags:
csch

O	U	E	N	C
		X	X	

acsch

O	U	E	N	C
			X	

Restrictions:
csch:



acsch - the domain of real numbers.

sech (asech)

Function Performed:

$$R = \frac{2}{(e^{X} + e^{-X})}$$

$$R' = ln\left(\frac{1 + \sqrt{1 - x^2}}{x}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
sech

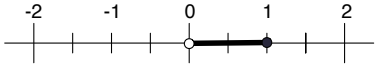
O	U	E	N	C
			X	

asech

O	U	E	N	C
		E	X	

Restrictions:
sech - the domain of real numbers.

asech:



coth (acoth)

Function Performed:

$$R = \frac{cosh(x)}{sinh(x)} = \frac{e^{2x} + 1}{e^{2x} - 1}$$

$$R' = \frac{1}{2} ln\left(\frac{X+1}{X-1}\right)$$

Consumed:
X

Results in:
X

Alt Function:
YES

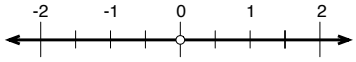
Flags:
coth

O	U	E	N	C
		X	X	

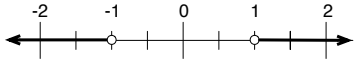
acoth

O	U	E	N	C
		X	X	

Restrictions:
coth



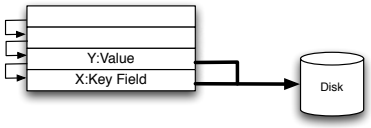
acoth:



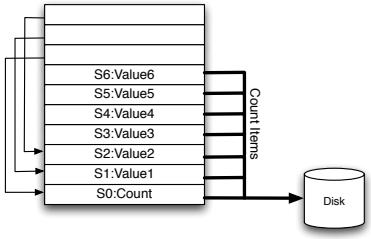
y->[x] (s[]->)

Function Performed:

R



R'



Consumed:

X (potential the whole stack)

Results in:

None

Alt Function:

YES

Flags:

y->[x]

O	U	E	N	C
X			X	

s[]->

O	U	E	N	C
X		E	X	

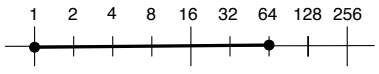
Restrictions:

y->[x]

None

s[]->

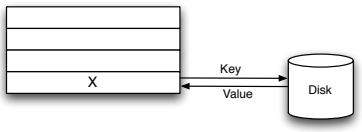
X Integer and:



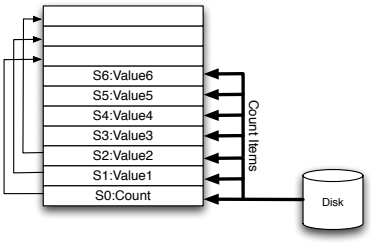
[x]->x (->s[x])

Function Performed:

R



R'



Consumed:

X

Results in:

Y Moved to storage and X

Alt Function:

YES

Flags:

[x]->x

O	U	E	N	C
X			X	

->s[x]

O	U	E	N	C
X		E	X	

Restrictions:

[x]->x

None

->s[x]

Stack register must exist

Count

Function Performed:

R = Counts from bottom of stack to last non-0 stack value

Consumed:

X, and X additional stack items.

Results in:

X

Alt Function:

NO

Flags:

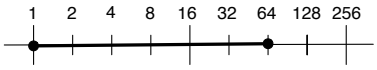
sum

O	U	E	N	C
X		X		

Restrictions:

sum

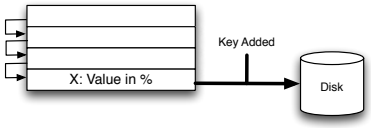
X - Must be integer and



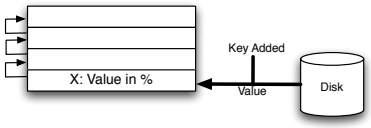
x->% (%->x)

Function Performed:

R



R’



Consumed:

X

Results in:

X

Alt Function:

YES

Flags:

x->%

O	U	E	N	C
X			X	

s[]->

O	U	E	N	C
X		E	X	

Restrictions:

y->[x]

None

s[]->

% must exist in storage

mean (mdian)

Function Performed:

mean (simple average)

$$R = \frac{1}{x}\sum_{i=1}^{X} S[i]$$

mdian (median)

Sort. report mid point. When X is even, report average of two midpoints.

Consumed:

X, and X additional stack items.

Results in:

X

Alt Function:

YES

Flags:

mean

O	U	E	N	C
X		X	X	

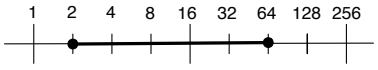
mdian

O	U	E	N	C
X		X	X	

Restrictions:

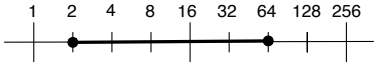
mean

X - Must be integer and



mdian

X - Must be integer and



stdev (psdev)

Function Performed:

stdev Sample Deviation

$$R = \sqrt{\frac{1}{X\text{-}1}\sum_{i=1}^{X}\left(S[i] - \frac{1}{x}\sum_{i=1}^{X} S[i]\right)^2}$$

psdev Population Deviation

$$R' = \sqrt{\frac{1}{x}\sum_{i=1}^{X}\left(S[i] - \frac{1}{x}\sum_{i=1}^{X} S[i]\right)^2}$$

Consumed:

X, and X additional stack items.

Results in:

X

Alt Function:

YES

Flags:

stdev

O	U	E	N	C
X		X		

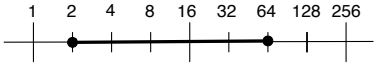
psdev

O	U	E	N	C
X		X	X	

Restrictions:

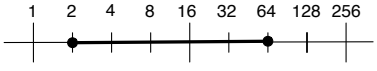
stdev

X - Must be integer and



psdev

X - Must be integer and



∑S[] (∏S[])

Function Performed:

sum Simple Sum

$$R = \sum_{i=1}^{X} S[i]$$

prdct Simple Product

$$R' = \prod_{i=1}^{x} S[i]$$

Consumed:

X, and X additional stack items.

Results in:

X

Alt Function:

YES

Flags:

sum

O	U	E	N	C
X		X	X	

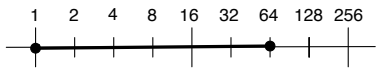
prdct

O	U	E	N	C
X		X	X	

Restrictions:

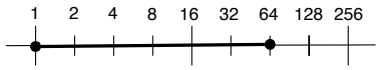
sum

X - Must be integer and



prdct

X - Must be integer and



∑S^2

Function Performed:

$$R = \sum_{i=1}^{X} S[i]^2$$

Consumed:

X, and X additional stack items.

Results in:

X

Alt Function:

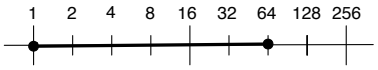
NO

Flags:

O	U	E	N	C
X		X		

Restrictions:

X - Must be integer and



Count

Function Performed:

R = Counts from bottom of stack to last non-0 stack value

Consumed:

None

Results in:

X

Alt Function:

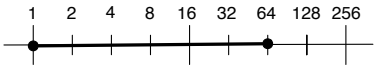
NO

Flags:

O	U	E	N	C
X		X		

Restrictions:

X - Must be integer and



yChsX

Function Performed:

Binomial Coefficient

$$R = \binom{y}{x} = \frac{y!}{x! \cdot (y - x)!} = \frac{y^{\underline{x}}}{x!}$$

Consumed:

X, Y

Results in:

X

Alt Function:

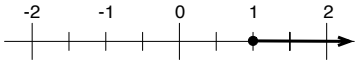
NO

Flags:

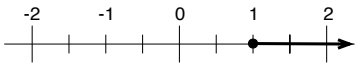
O	U	E	N	C
X		X		

Restrictions:

X - Must be integer and greater than 0:

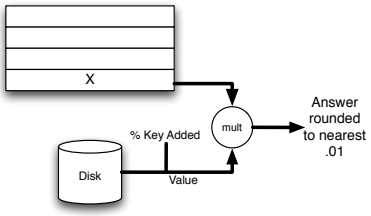


Y - Must be integer and greater than or equal to X



X * %

Function Performed:



Consumed:
X

Results in:
X

RESULT IS ROUNDED TO NEAREST 0.01.

Alt Function:
NO

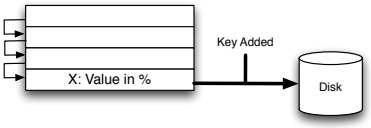
Flags:
sum

O	U	E	N	C
X		X		

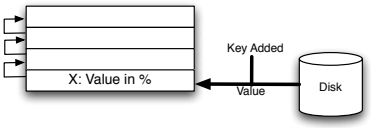
Restrictions:
% Storage register must exist.

x->% (%->x)

Function Performed:
R



R’



Consumed:
X

Results in:
X

Alt Function:
YES

Flags:
x->%

O	U	E	N	C
X			X	

s[]->

O	U	E	N	C
X		E	X	

Restrictions:

y->[x]
None

s[]->
% must exist in storage

%

Function Performed:
R = X / 100

Consumed:
X

Results in:
X

Alt Function:
YES

Flags:

O	U	E	N	C
	X			

Restrictions:
None

Δ%

Function Performed:

$$R = \frac{y - x}{x} \cdot 100\%$$

Consumed:
X

Results in:
X

RESULT IS ROUNDED TO NEAREST 0.01.

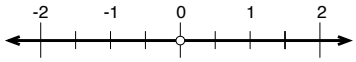
Alt Function:
NO

Flags:

O	U	E	N	C
X	X	X	X	

Restrictions:

X:



Y: The domain of real numbers.

fVal

Function Performed:
Future Value

$$R = PV \cdot (1 + i)^t$$

$$R = z \cdot \left(1 + \frac{y}{100}\right)^x$$

Z: = PV (present Value)
Y: = ***i*** (interest/cycle)
X: = ***t*** (cycles)

Consumed:
X, Y, Z

Results in:
X

RESULT IS ROUNDED TO NEAREST 0.01.

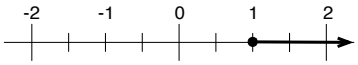
Alt Function:
YES

Flags:

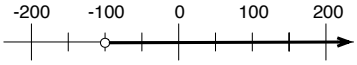
O	U	E	N	C
X	X	X	X	

Restrictions:

X:



Y:



Z: The domain of real numbers.

pVal

Function Performed:
Present Value

$$R = \frac{FV}{(1+i)^t}$$

$$R = z \cdot \left(1+\frac{y}{100}\right)^{-x}$$

Z: = PV (present Value)
Y: = ***i*** (interest/cycle)
X: = ***t*** (cycles)

Consumed:
X, Y, Z

Results in:
X

RESULT IS ROUNDED TO NEAREST 0.01.

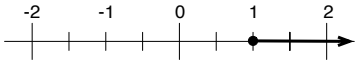
Alt Function:
YES

Flags:

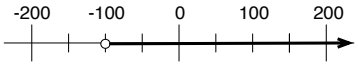
O	U	E	N	C
X	X	X	X	

Restrictions:

X:



Y:



Z: The domain of real numbers.

pi

Function Performed:
R = M_PI

R =
3.14159265358979323846264
338327950288

Consumed:
None

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C

Restrictions:
None

e

Function Performed:
R = M_E

R =
2.71828182845904523536028
747135266250

Consumed:
None

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C

Restrictions:
None

c

Function Performed:
Speed of light

R =299792458.0

Units: m / s

Value is exact by definition

Consumed:
None

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C

Restrictions:
None

h

Function Performed:
Planck Constant in

R =6.626068 × 10-34

Units: m^2 kg / s

Consumed:
None

Results in:
X

Alt Function:
NO

Flags:

O	U	E	N	C

Restrictions:
None

∞ (-∞)

Function Performed:
Infinity

R = ∞

R' = -∞

Consumed:
None

Results in:
X

Alt Function:
YES

Flags:

O	U	E	N	C
x				

Restrictions:
None

NAN (-NAN)

Function Performed:
Not A Number

R = NAN

R' = -NAN

Consumed:
None

Results in:
X

Alt Function:
YES

Flags:

O	U	E	N	C

Restrictions:
None