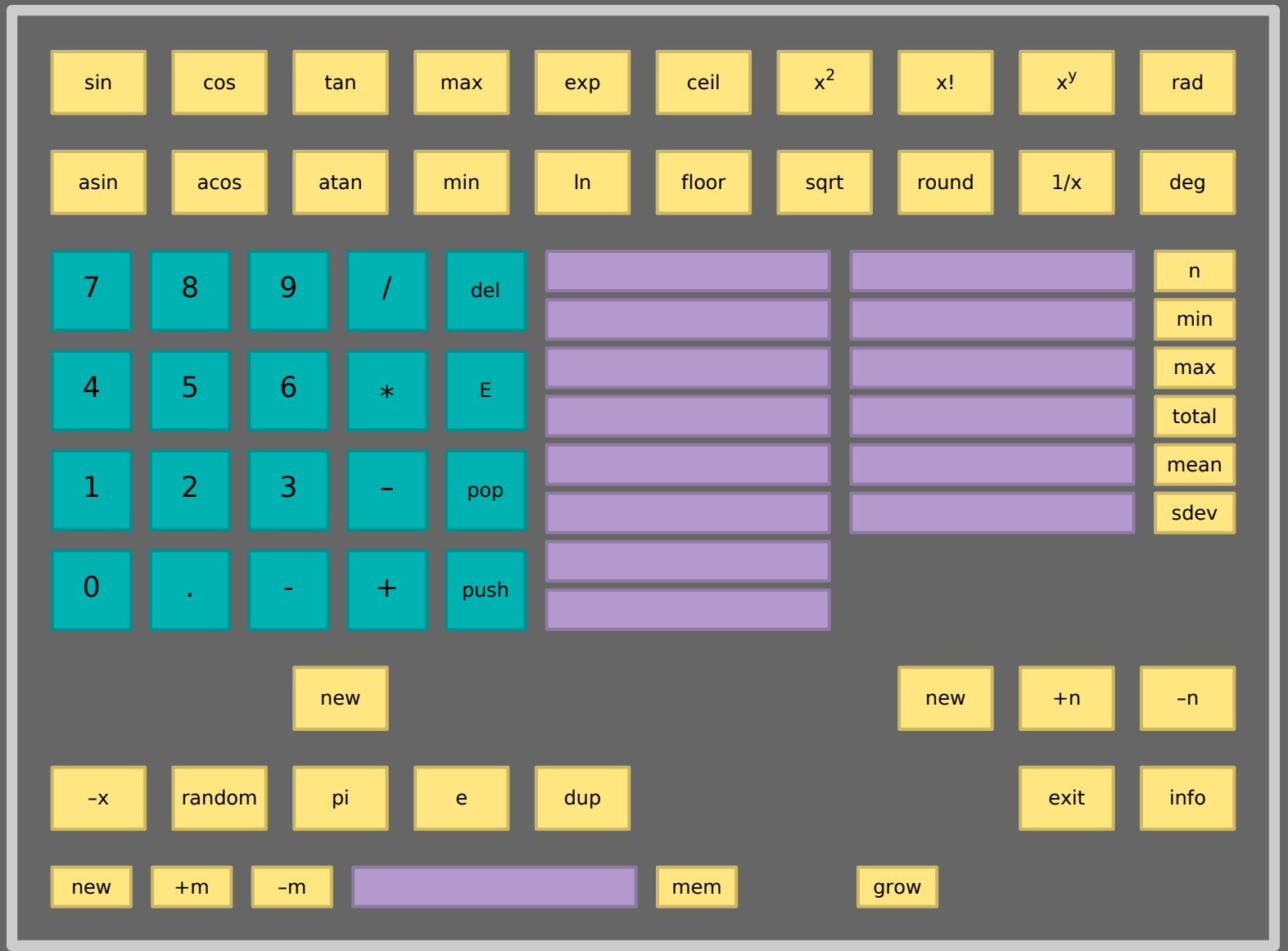

sin
cos
tan
max
exp
ceil
x2
x!
xy
rad
asin
acos
atan
min
ln
floor
sqrt
round
1/x
deg
7
8
9
/
del
4
5
6
*
E
1
2
3
–
pop
0
.
-
+
push
n
min
max
total
mean
sdev
new
new
+n
–n
–x
random
pi
e
dup
exit
info
new
+m
–m
mem
grow

The Calculator

This calculator is stack based, which means that one enters values and invokes an action that acts on the value(s) last entered. Subtracting 10 from 20 using (**-**) for instance comes down to clicking:

10 in 20 –

while calculating a sinus (**sin**) results from entering:

.89 sin

The left column of fields (numbers) shows the Stack. One uses **push** to push a value on the stack and **pop** to remove a value. Clicking **new** removes them all and the **del** button can be used to undo the last entered digit. When a dyadic operation is applied, the top value is used as y. The **grow** key toggles between two different visualizations of the stack.

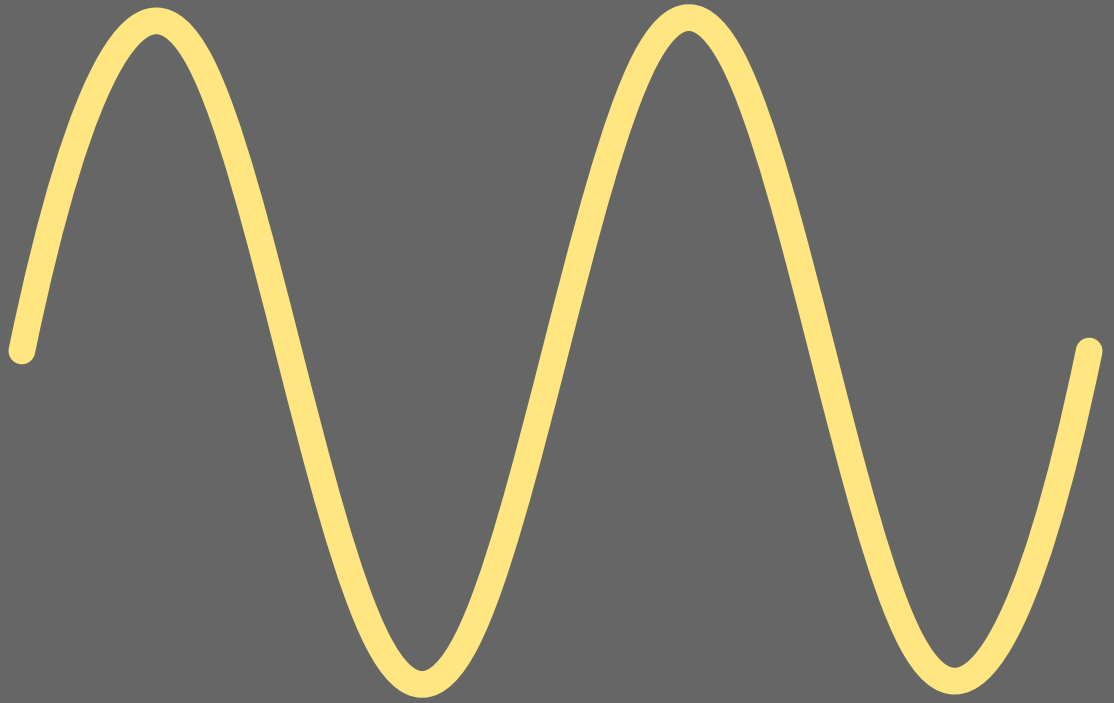
The stack is considerably larger than the screen representation suggests. In the rare occasion that one encounters the message exhausted, the amount of stack entries already has totaled far beyond 50 and one probably already has forgotten what the values first entered represent.

The right column of fields reports the statistic calculations. By clicking on the tag, one pushes the value on the Stack. The lower buttons are used to reset (**new**), enter (**+**) and remove (**-**) values to be taken into account when calculating those statistics.

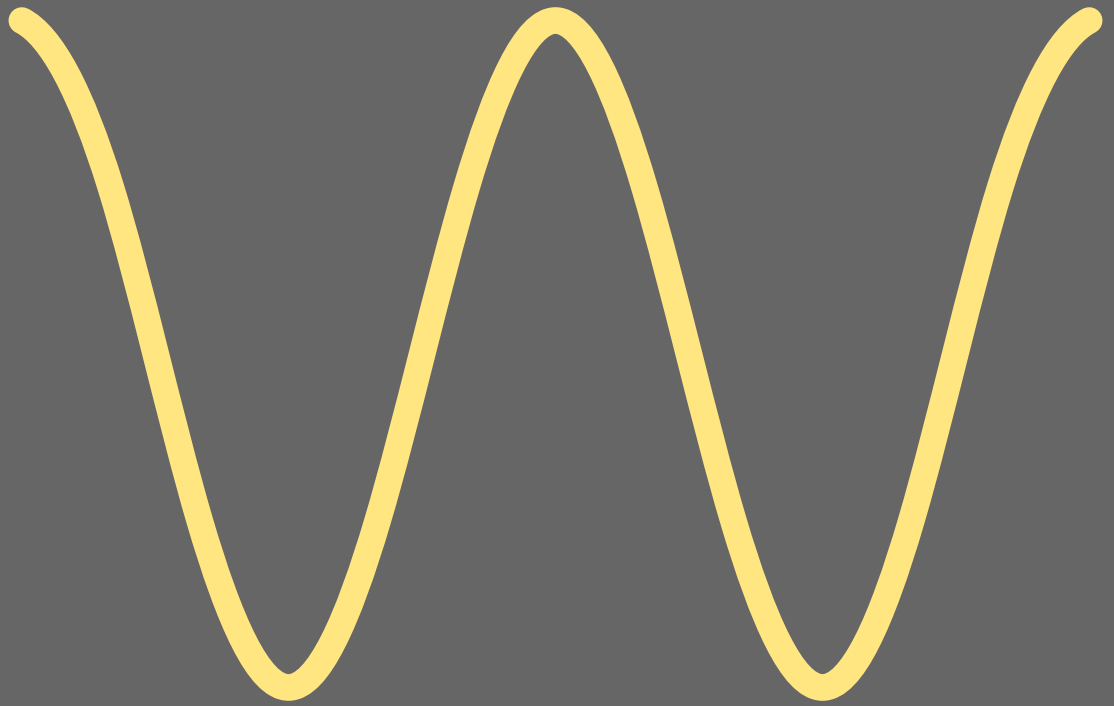
This document is produced by ConTEXt, a macro package written in TEX. The graphics are METAPOST graphics. The graphics, the PDF objects and the form fields as well as JavaScript code were generated and inserted at run time. Originally we used PDFTEX and MkII to process this document but this one is done by LuaTEX and MkIV. We kept the design and code original so that it reflects how things were done (for readability we updated some TEX definitions).

Hans Hagen, PRAGMA ADE, ConTEXt 18/2/1998–25/9/2018

mailto:pragma@xs4all.nl

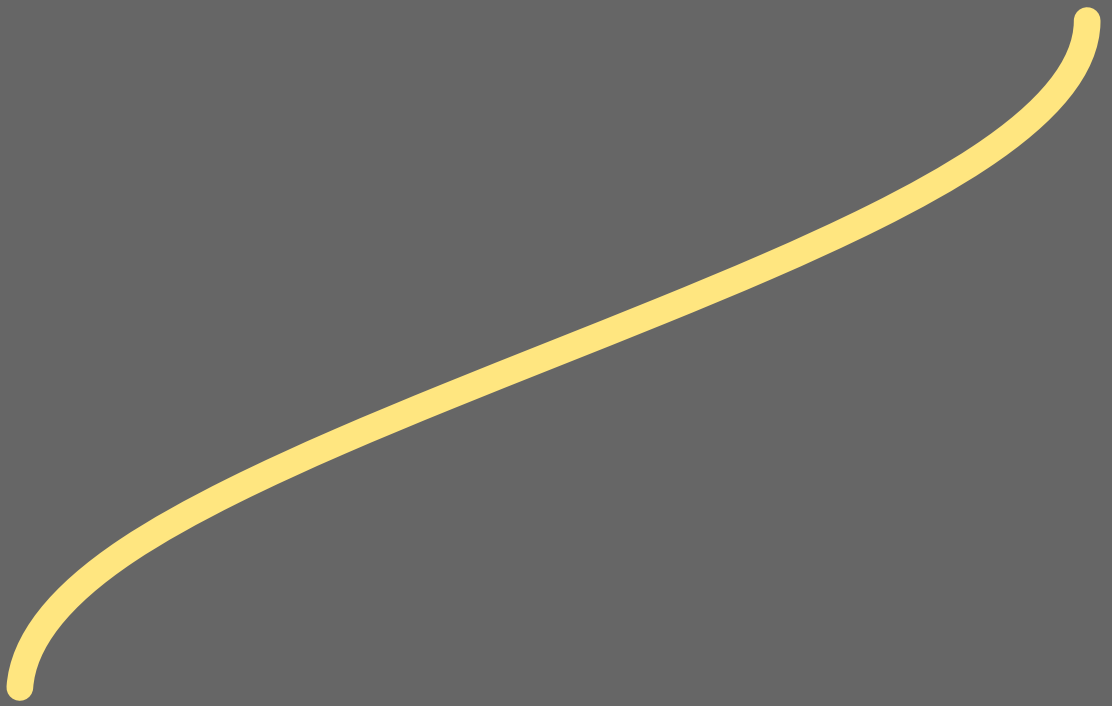


Calculate the sine of the topmost stack entry.

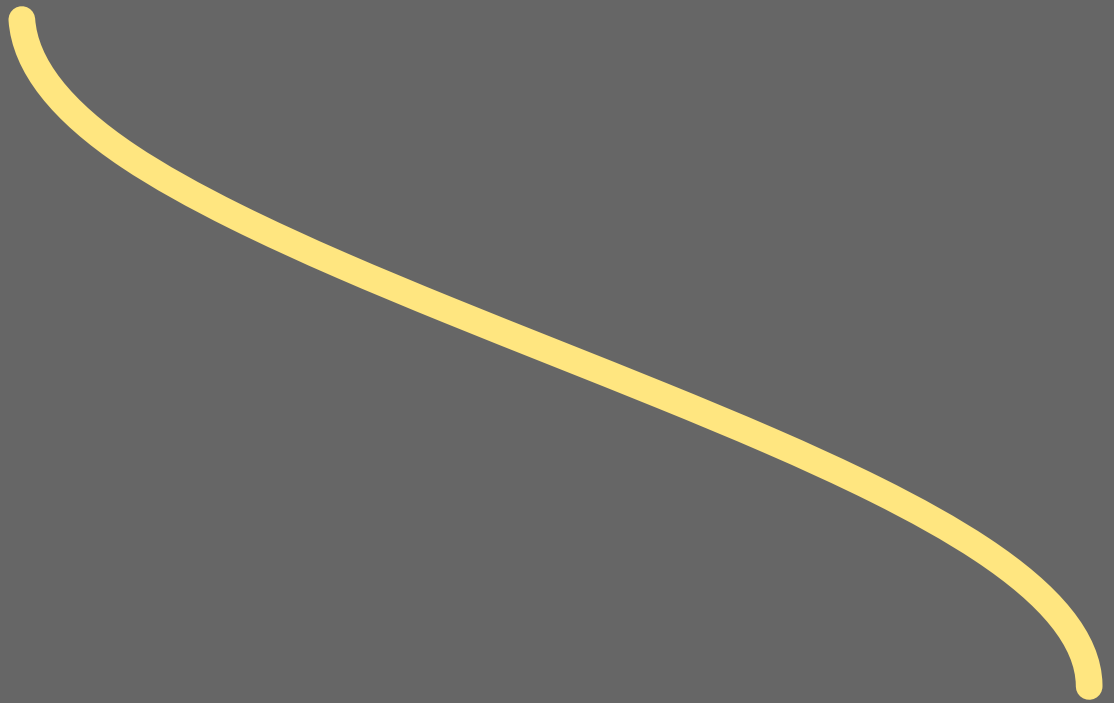


Calculate the cosine of the topmost stack entry.

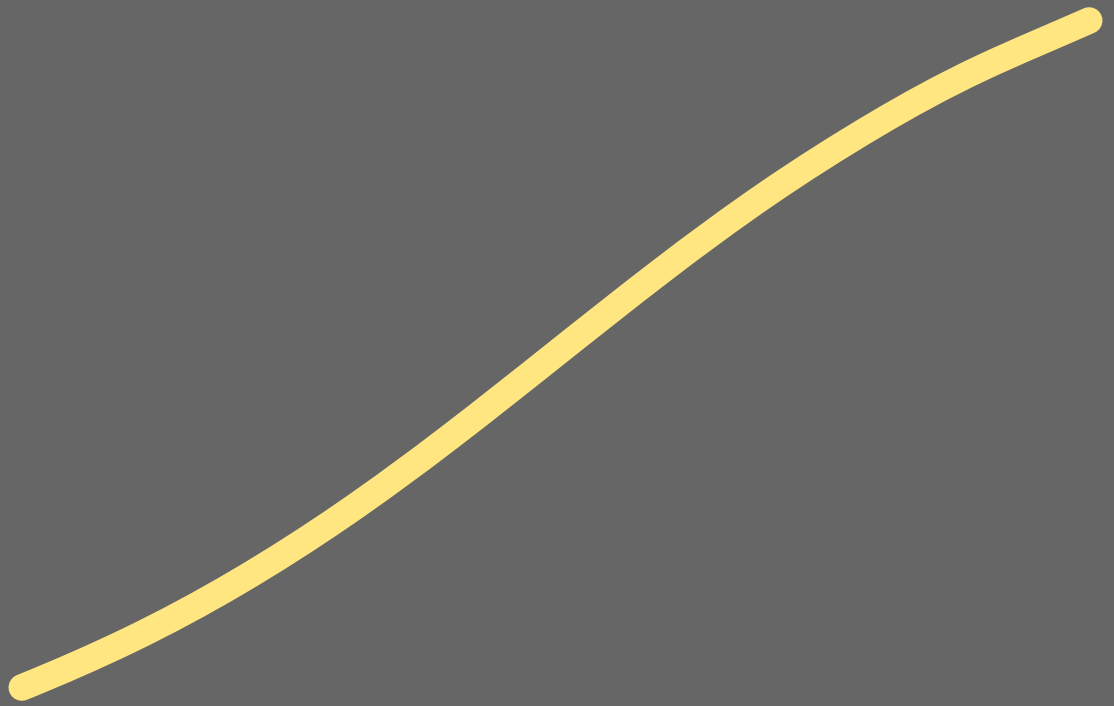
Calculate the tangent of the topmost stack entry.



Calculate the arcsine of the topmost stack entry.

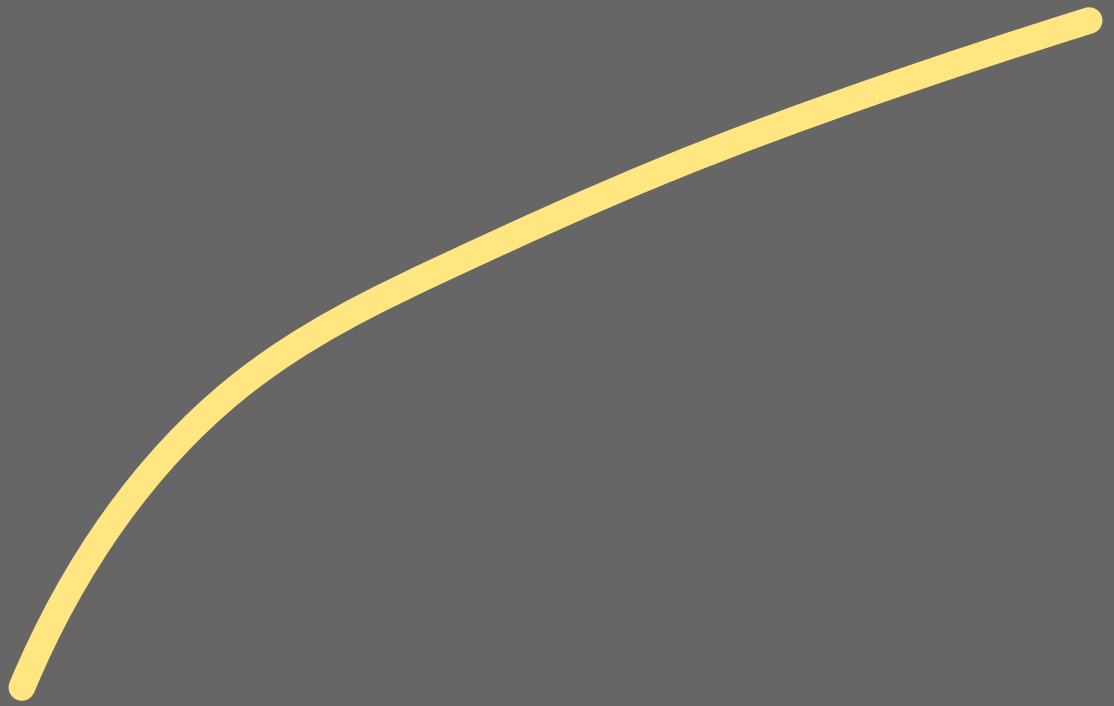


Calculate the arccosine of the topmost stack entry.



Calculate the arctangent of the topmost stack entry.

Calculate the square of the topmost stack entry.



Calculate the square root of the topmost stack entry.

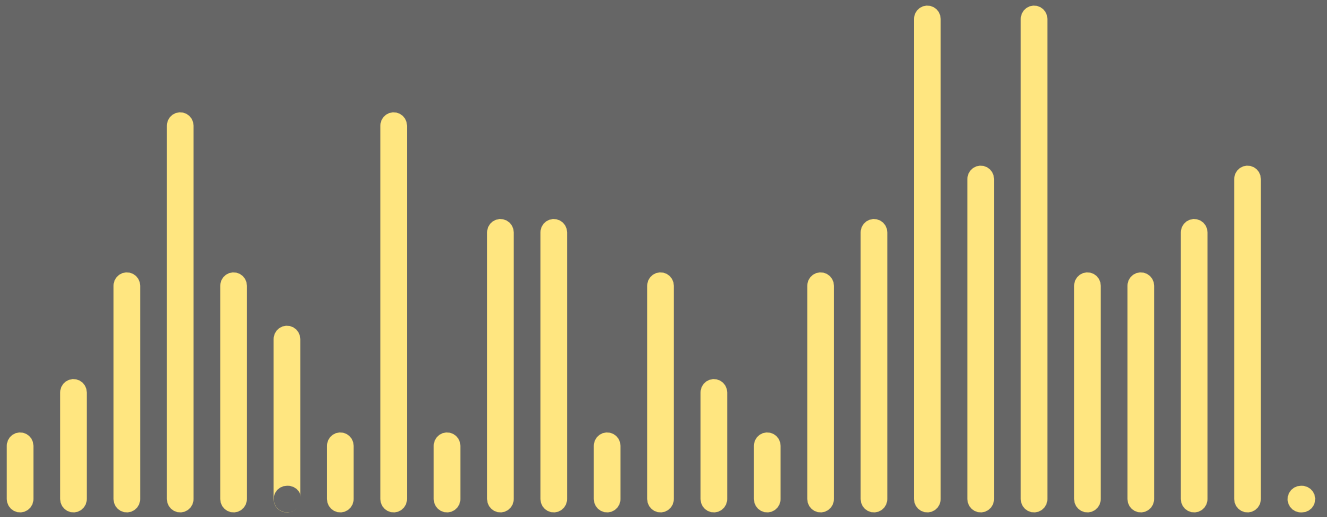
Calculate the exponential function of the topmost stack entry.

Calculate the natural logaritm of the topmost stack entry.

Calculate x^y where y is the topmost stack entry.

Calculate 1/x using the topmost stack entry.

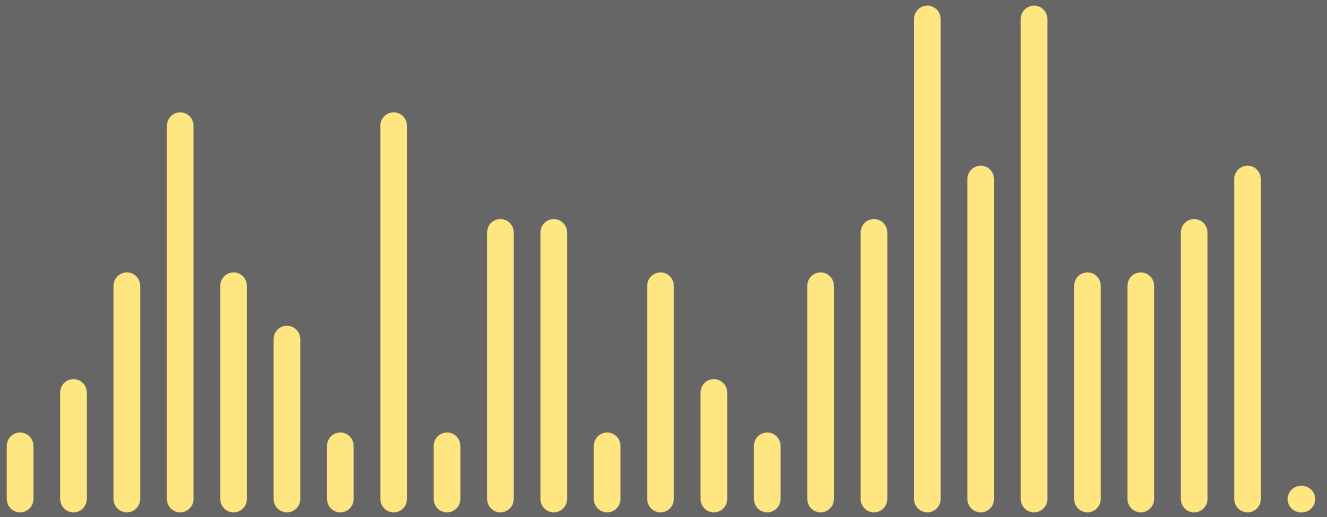
Add an observation to the statistics.



Remove an observation from the statistics.

Reset the statistics.

Push the number of observations to the stack.

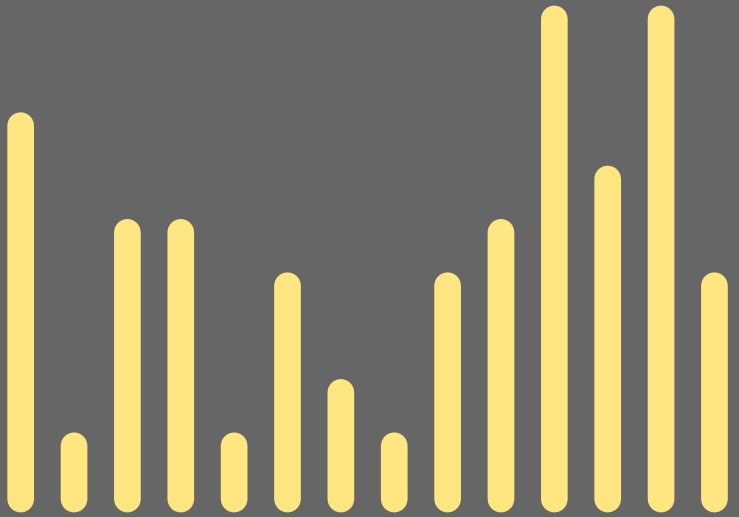


Push the sum of encountered values to the stack.

Push the lowest encountered value to the stack.

Push the highest encountered value to the stack.

Push the mean value to the stack.



Push the standard deviation to the stack.

Negate the topmost stack entry.

Set the topmost stack entry to the next integer.

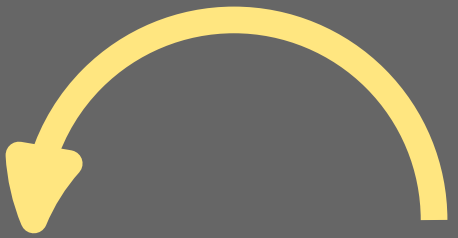
Set the topmost stack entry to the previous integer.

Set the topmost stack entry to the nearest integer.

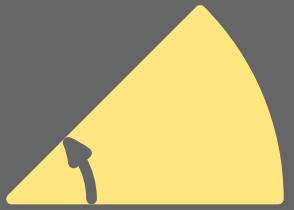
Take the minumum of the two topmost stack entries.

Take the maximum of the two topmost stack entries.

Push 2.71828182845905 onto the stack.

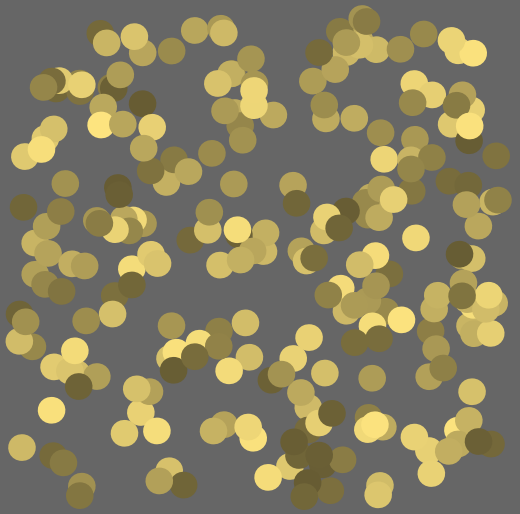


Push 3.14159265358979 onto the stack.



Convert radians into degrees.

Convert degrees into radians.



Generate a random number in the range 0–1.

Add a digit 0 to the current stack entry.

Add a digit 1 to the current stack entry.

Add a digit 2 to the current stack entry.

Add a digit 3 to the current stack entry.

Add a digit 4 to the current stack entry.

Add a digit 5 to the current stack entry.

Add a digit 6 to the current stack entry.

Add a digit 7 to the current stack entry.

Add a digit 8 to the current stack entry.

Add a digit 9 to the current stack entry.

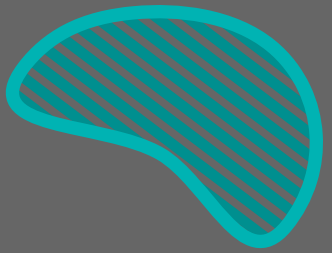
Calculate the recursive multiplication of n, n–1, n–2, etc.

Add a sign to the current stack entry.

Add a period to the current stack entry.

Start setting the exponent part of the current stack entry.

Delete the last entered digit of the current stack entry.



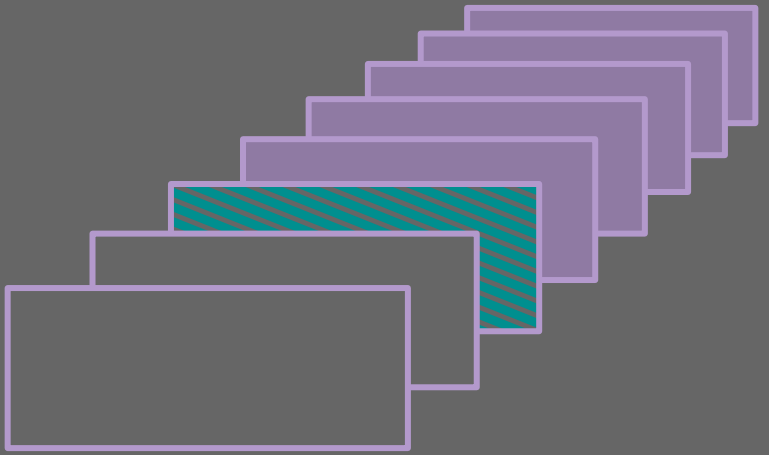
Erase the memory buffer.

Add to the memory buffer.

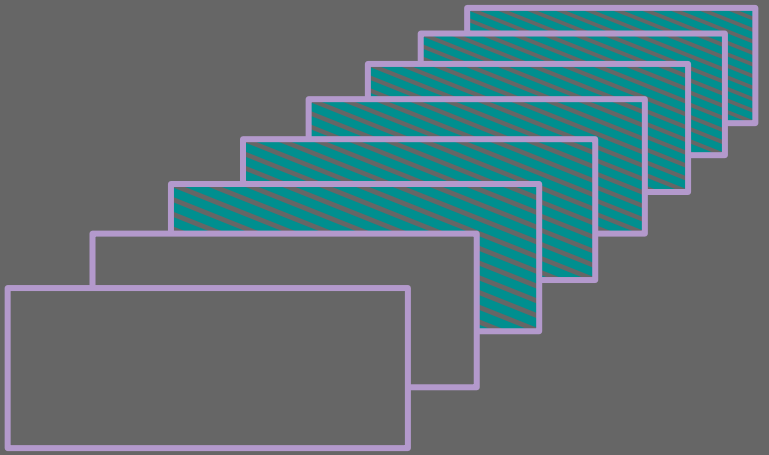
Substract from the memory buffer.

Copy the memory buffer to the stack.

Push a new entry to the stack.

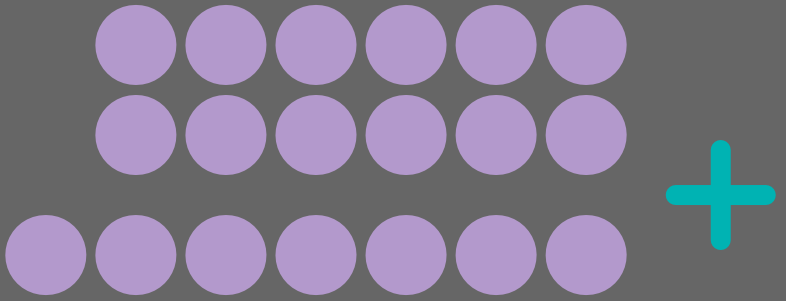


Remove the topmost entry from the stack.

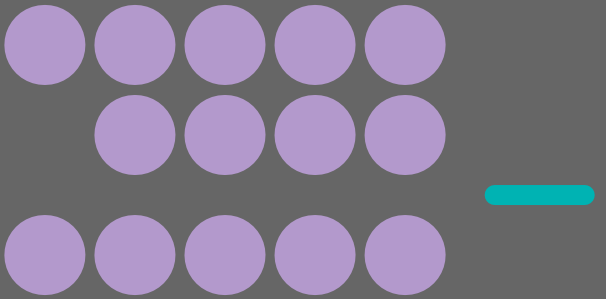


Erase the whole stack.

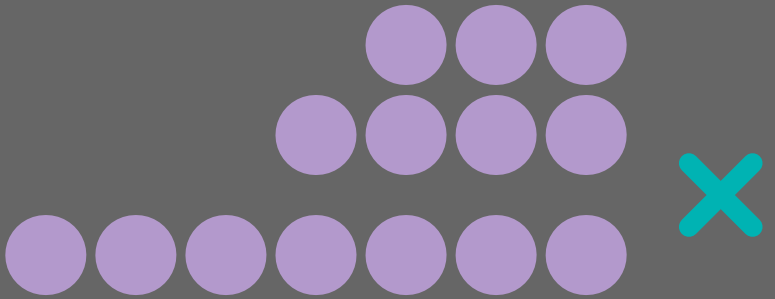
Duplicate the topmost stack entry.



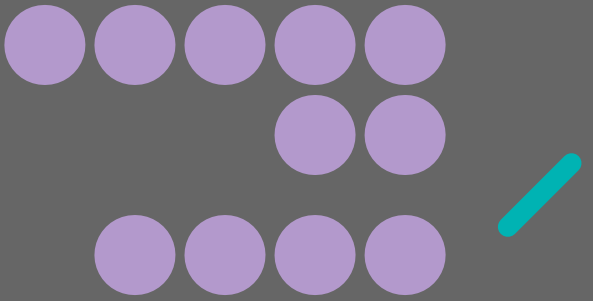
Add the two topmost stack entries.



Subtract the topmost stack entry from the one below.

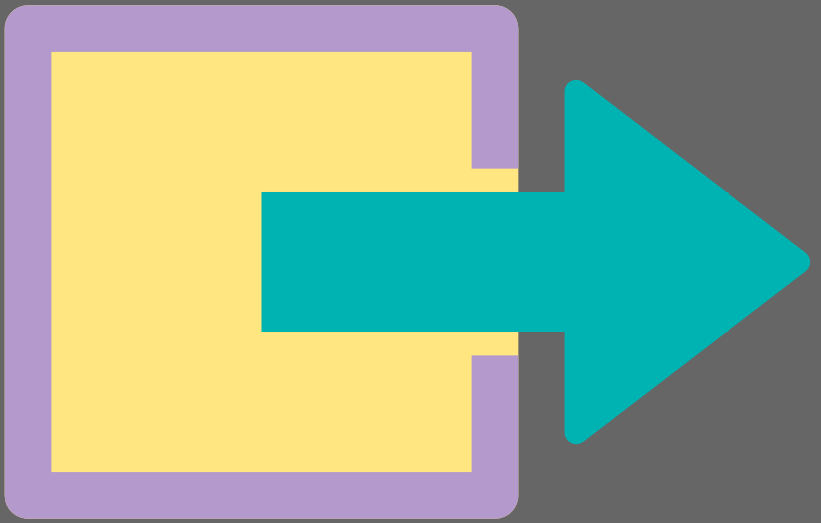


Multiply the two topmost stack entries.



Divide the pre-last stack entry by the topmost one.

Toggle grow mode, another way of stacking.



Close this document.