

**XCALC Ver 1.2**



## **XCALC Help Index**

### **In Short**

[About XCALC](#)

### **What's new to Version 1.2**

[Version1.2 news](#)

### **Operation**

[RPN calculator](#)

### **Modes**

[Maths and Display settings](#)

### **Keyboard**

[Keyboard function mapping](#)

### **Mouse**

[Mouse operations](#)

### **Setup**

[Configuring XCALC](#)

### **Mathematical Operation**

[Complex numbers, and accuracy](#)

### **Author and such...**

[Who wrote this program](#)

## **XCALC Ver 1.2**

If you double-click on this logo, click the Help button or press F1, you will bring up the *Windows Help* for XCALC.

[Index](#)



## **XCALC.EXE**

Version 1.2, August 1993

Bernt Ribbum

bernt@hsr.no

[Index](#)

## What's new to Version 1.2

New to Version 1.2 (from 1.1) are the following items:

A new Keyboard display for complete mouse operation

Therefore no more KeyHelp

All buttons are now real pushbuttons

Inactive buttons are now greyed out / not shown

You can resize the window with the mouse

A few mathematical bugs have been corrected

The memory popup window show more digits

Error messages can be removed by clicking on the message itself (or pressing any key)

### See also:

What's new to Version 1.1

"KeyHelp" was the separate application that could be run alongside XCALC to give visual indication of the different keystrokes used to access the available functions. You will now have to use this help file to find the keystrokes.

There were some complaints to Version 1.1 of XCALC that inactive buttons still *looked* active. This has now been changed, so there is visual indication that e.g. when you select one of the binary formats (Hex, Oct, Bin), the decimal fixation and trig mode buttons are not operative.

Up to 14 significant digits are shown in the decimal/complex modes. Rounding errors may therefore sometimes be visible.

## What's new to Version 1.1

New to Version 1.1 (from 1.0) are the following items:

Clipboard Copy/Paste

Separate VGA/Super VGA Borland Dialogue Boxes (looks prettier)

Option to make XCALC topmost

Stack size can be as big as 8

Can customise the display to show or hide the various panes

Extra popup window with memory contents

Memory can be saved between sessions (to file XCALC.DAT)

Corrected overflow error with +, -, \* and / operation

### See also:

What's new to Version 1.2



## Keyboard Display

The biggest difference between Ver 1.1 and 1.2 is the inclusion of a "keyboard" that can be operated with the mouse. This gives access to all functions. The "keyboard" will change according to which *radix* you select XCALC to operate in.

Most of the keys should be self-explanatory, with the possible exception of the following:

*Decimal/Complex:*

**"x+iy"** The real parts of registers y and x are combined to a complex number.

*Complex:*

**"j"** Start entering imaginary part of the number.

Note that the text on the keys is drawn using a combination of the standard scalable Arial and Symbol fonts; the latter includes most of the common symbols like " $\sqrt{\quad}$ " and " $\pi$ ", but the combination does not always look exactly right. The keyboard can look especially ugly below 800 by 600 resolution.

**See also:**

[Configuring XCALC](#)

From Version 3.1 of *Windows*, any window can be made topmost, i.e., it will never be obscured by other windows and is thus always visible.

## **XCALC in short**

XCALC is a highly configurable, "extended" calculator for *Windows* 3.1, that you can use for simple calculations using decimal or complex, hexadecimal, octal or binary numbers (signed 2's complement). The program uses RPN (Reverse Polish Notation), similar to Hewlett Packard handhelds. You can choose from anything between a small window consisting of a single number display only (which you can float above other programs) and a full-fledged big display with all buttons, bells and whistles.

The user interface is designed for ease of use, and you can use the keyboard or mouse for all operations as you please. To find **123+456**, for example, key in "**123 <Enter> 456 +**". The weakness with using the keyboard is of course the lack of square root keys and such on the PC, but the keyboard interface is definitely faster than the mouse. If you can't remember the keys for the various mathematical operations, you can always use the keyboard display for those. You will also find memory store/recall functions, interaction with the clipboard, setting of various modes such as Decimal/Complex/Hex/Octal/Binary, and "Fix" or "Float" decimal settings.

XCALC will remember its settings between sessions; in its .INI file are kept the various configurable items, and also the screen position.

### **See also:**

[RPN operation](#)

[XCALC buttons](#)

[Keyboard display](#)

[Configuring XCALC](#)

## **XCALC Buttons**

On the XCALC main window there are now plenty of buttons that can be clicked, arranged in groups that can individually be switched on or off (see [configuration](#)):

[Keyboard group](#)

[Radix group](#)

[Display mode group](#)

[Fix / General group](#)

### **See also:**

[Configuring XCALC](#)

[Modes](#)

Those buttons are the equivalent of the standard calculator keyboard with the digits and mathematical functions. See [keyboard](#)

With those buttons you can select the number system (radix) you want to use - Decimal, Complex, Hexadecimal, Octal, Binary.

Here you can select how the *decimal* and *complex* numbers are displayed (with or without an exponent), and also the trigonometric mode (degree or radian measure).

This group gives you mouse access to the *decimal* setting (i.e., how many decimals to display), and also the three pushbuttons Configure, Help and Exit.



## RPN Calculator

In short, an RPN calculator operates using a stack, into which you push numbers for subsequent maths operations. This means you enter the numbers first, then you calculate.

**Example:** to calculate "3+4" using an RPN calculator, you enter the following:

**3**  
**Enter**  
**4**  
**+**

The RPN calculator is virtually modeless, and you do not have to worry about how many nested levels of parentheses you have keyed in (you can't). The only mode (apart from the mathematical modes) is the Stack Lift: this is an On/Off mode that tells the calculator whether or not the next number keyed in should overwrite the X register. Stack Lift is *on* at all times except when Enter or BackSpace was just pressed.

Standard maths functions that operate on one number only are called without pressing enter:

**Example:** to calculate "sin (3)" on the RPN calculator, press:

**3**  
**s** (or click the Sin button)

**See also:**

[Last X register](#)  
[Maths modes](#)  
[Keyboard function mapping](#)  
[Mouse operations](#)  
[XCALC buttons](#)

## Last X:

The *LastX* register is an invisible one that will store the last value that was used from the X register in a mathematical operation. The value is recalled by pressing "l" (lowercase ell) or clicking the LastX button.

### Example:

Enter **3** Enter **4 +**

This will yield the answer of **7** in the X register, and the Last X register will contain the last value from the X register, namely, **4**.

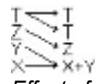
This can be used as a primitive *undo* function, provided you know which operation you did last, and that there is an inverse operation. For example, immediately following the example above, you can enter:

**l-l** (ell, minus, ell)

This will restore the stack to the state it had just prior to the addition. (The common *Alt-Backspace* undo is not implemented.)

RPN is short for "Reverse Polish Notation"

If you want to be finicky, the RPN stack is not really a stack, since the RPN stack never becomes empty..!  
Also, the top register (T) is always copied down to the register immediately below it following a binary  
maths operation, instead of inserting a zero in T. Example:



*Effect of addition.*

The "stack" registers (default 4) are named:

T (top)

:

Z

Y

X (the 'active' register)

An additional register is Last X

Use the keyboard to enter numbers and operators.

Use the "Enter" or "Return" key on the keyboard.

The BackSpace key will clear the last digit if you are entering a number; otherwise it will clear the X register and disable the stack lift.



See Function Keys

## Modes

There are 5 modes of operation for XCALC, divided into two distinct groups, floating point and 32-bit signed integer:

**Floating Point:** [Decimal](#)  
[Complex](#)

**Integer:** [Hexadecimal](#)  
[Octal](#)  
[Binary](#)

For the **Floating Point** modes, various settings will be available:

### Fix/Float:

Using the [mouse](#) or the [keyboard](#), you can select the display of numbers using a fixed amount of digits behind the decimal point ([fixed](#)) or a variable amount ([float](#)). You can also choose between the [Fix](#), [Scientific](#) and [Engineering](#) modes.

### Trigonometric settings:

For Decimal (not Complex) operation, you can choose between the [Degree](#) or [Radian](#) angular measurements. The degree measure does not make much sense for complex numbers, so the only setting available for complex operation is the Radian measurement. (See [Complex Operation](#) for an explanation.) If you switch back to Decimal mode, your original setting is restored.

Normal mode, radix 10.

Real and Imaginary parts of the numbers, radix 10.

Radix 16, digits 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e and f. Treated as signed numbers.

Radix 8, digits 0,1,2,3,4,5,6 and 7. Treated as signed numbers.

Radix 2, digits 0 and 1. Treated as signed numbers.

There are 360 degrees in a full circle.



There are  $2\pi$  radians in a full circle.

Examples of "float" numbers:

-3.878455

456.6

50

The same numbers with "fix" 2 decimals:

-3.88

456.60

50.00

Fix: No forced exponent, e.g., **3.1416**.

Scientific: Always an exponent of 10, e.g., **-1.00 e+001**.

Engineering: Always an exponent of 10, which is always divisible by 3, e.g., **66.67 e+003**.

## Complex Number Calculations

The following explains some of the reasoning behind the operation of XCALC. If you are not familiar with complex numbers, go read a maths textbook.

I have chosen a complex-only solution for the floating-point routines in XCALC. This means that when you do decimal calculations, behind the scene quite a few invisible things happen (like addition of Imaginary parts of numbers). Normally, this goes without any problems, and you might appreciate the solution if you want to know, say, the square root of -1. If the imaginary part as the result of any calculation becomes non-zero, the mode is automatically set to Complex. In the [accuracy](#) section you can see some unfortunate effects of this, but it certainly makes the internal operation of XCALC a lot tidier!

Since complex numbers and radian angle measures are inseparable, there is no provision for "degrees" in the Complex mode. Euler's formulae simply do not make any sense unless radians are used.

There is also an ambiguity for many complex operations; the value returned by XCALC is correct, but only to a certain degree. Because of this, not all functions will have a true *inverse* with complex numbers. For example,  $\cos(1+i)$  will with XCALC result in  $0.83... + i0.98...$ , but the inverse operation (arccos) of this will yield  $1-i$ . [A result of this is that  $\arctan(\tan(x))$  often has no resemblance to  $x$  itself.]

**See also:**

[Accuracy](#)

## Accuracy

XCALC works (for floating-point modes) internally with Double Precision numbers. This accounts for an *approximate* accuracy of 15 decimal digits. This is, however, not precise, and certain operations may generate errors that accumulate to be larger than this. I will give two examples to show what to expect:

### Example 1:

Calculate the square root of the square of five, minus 5:

Press: **5 Q q 5 -**

The result is *not* zero, but approximately  $-8.88e-16$ . This is because the square root is not exact.

### Example 2:

Calculate minus 1 to the power of two:

Press: **1 m Enter 2 ^**

The result is complex! Again, the reason is inaccuracy in the computation, which is done using logarithms for the power function, with in this case a small "leakage" into the imaginary part. (Many handheld calculators can do this correctly; they operate internally with BCD numbers, and can thus always tell an integer from a non-integer).

Binary Coded Decimal



## Keyboard Function Mapping

The keyboard operation depends partly on the mode used; see the following topics.

**General Keys** (all modes)

**Decimal/Complex keys**

**Hexadecimal/Octal/Binary keys**

**See also:**

Mouse operations

## General Keys (always operating)

## Keyboard

<b>Key</b>	<b>Explanation</b>
Enter	Finish entering digits, push stack upwards;
+ - * /	These keys are for normal input and operation;
I (ell)	Recall last X;
[up], [down]	Roll up/Roll down stack.
[BackSpace]	BackSpace / Clear X;
Z, X	Clear stack;
Alt-0 - Alt-9	Set FIX mode to zero through nine decimals;
Alt-F	Set <b>F</b> LOAT mode (varying number of decimals);
Alt-I	Set normal (Fix/Float) mode;
Alt-S	Set <b>S</b> cientific mode;
Alt-E	Set <b>E</b> ngineering mode;
Alt-D	<b>D</b> ecimal;
Alt-C	<b>C</b> omplex;
Alt-H	<b>H</b> exadecimal;
Alt-O	<b>O</b> ctal;
Alt-B	<b>B</b> inary;
Alt-G	Set <b>D</b> egree mode;
Alt-R	Set <b>R</b> adian mode;
Alt-N	Bring up configuration box;
Alt-X, Alt-F4	<b>E</b> xit;
Ctrl-Z, Ctrl-N	Toggle display of <u>memory popup</u> window;
Ctrl-C/Ctrl-Ins	<b>C</b> opy X (as text and bitmap) to the <u>clipboard</u> ;
Ctrl-V/Shift-Ins	<b>P</b> aste (text) from the <u>clipboard</u> into X;
Ctrl - 0-9	[ <i>Numeric keypad</i> ] Store to <u>memory register</u> <i>n</i> ;
Alt - 0-9	[ <i>Numeric keypad</i> ] Recall from memory register <i>n</i> ;
Ctrl-Alt - 0-9	[ <i>Numeric keypad</i> ] Clear memory register <i>n</i> ;
F1	Help;
Esc	Minimise window;

## Decimal/Complex mode Keys

## Keyboard

<b>Key</b>	<b>Explanation</b>
e	Enter exponent;
p	$\pi$ ;
s c t S C T	<b>S</b> ine, <b>C</b> osine and <b>T</b> angent, respectively; Arc <b>S</b> ine, Arc <b>C</b> osine and Arc <b>T</b> angent;
q Q	<b>S</b> quare root and <b>s</b> quare, respectively;
L E G N	Natural logarithm ( <b>ln</b> ) and exponential ( <b>e<sup>x</sup></b> ); Base-10 logarithm ( <b>log</b> ) and antilog ( <b>ten</b> , <b>10<sup>x</sup></b> );
^ &	Power ( <b>y<sup>x</sup></b> ) and root ( <b>y<sup>1/x</sup></b> );
	Absolute value ( <b> x </b> );
!	Factorial ( <b>x!</b> ) of the <i>integer real</i> part of <b>x</b> ;
%	Percent ( <b>x ← y*x/100</b> , <b>y</b> unchanged)
\	Reciprocal ( <b>1/x</b> );
i, j	Combine the real parts of <b>y</b> and <b>x</b> into <b>y + ix</b> . Switch to complex mode if in decimal mode;
[Left], [Right]	Set <b>FIX/FLOAT</b> modes;
' (single quote)	Complex conjugate (Complex only);
"	Swap real and imaginary part (Complex only);
, (comma)	Finish entering real part, start entering imaginary part of number (Complex only).

## Hexadecimal/Octal/Binary mode keys

## Keyboard

<b>Key</b>	<b>Explanation</b>
&   ~ ^	And, Or, Not and Xor;
%	Remainder
[Left], [Right]	Shift 1 bit Left/Right;
Shift-[Lft], [Rgt]	Rotate 1 bit Left/Right;

## Clipboard Functions

XCALC can both copy to, and paste from, the *Windows* Clipboard using the standard Copy/Paste keys (the operation is akin to memory store/recall: numbers are copied from/to the X register, and the stack is pushed if StackLift is *on*).

Using the mouse, you can bypass the stack operation and copy directly to/from any visible stack register by clicking on the display itself:

**Left Double-Click:** Copy to clipboard  
**Right Double-Click:** Paste from clipboard

The text is copied verbatim from the display, so if you use "fixed point" display, only the visible digits are copied. Blanks are removed. A paste operation is successful if a textual representation (ANSI or OEM text) of a number valid for the current radix setting is the first non-whitespace content of the clipboard (a maximum of 34 digits will be read). If a paste cannot be done (if there is no text on the clipboard, or if the text is non-numerical), no operation takes place at all. Successful copy and paste operations are indicated briefly on the display.

Note that a copy operation will also transfer a bitmap representation of the display to the clipboard. I can't see why anyone in their right mind would use this, but it's there anyway.

**Copy:** Ctrl-C or Ctrl-Ins  
**Paste:** Ctrl-V or Shift-Ins

There are 10 memory registers, numbered 0 through 9. Each can hold a number in any format. See [Memory Registers](#).

## Mouse Operations

I don't encourage operating a calculator with a mouse. Nevertheless, due to countless requests the XCALC display can now be made chock-a-bloc full of mouse-friendly buttons. Apart from clicking on all those function keys, you can also do the following operations with the mouse in XCALC:

- The standard move and system menu operations;
- Click on the buttons for various modes;
- Set the Fixed/Float display by clicking on the small digits in the Fix/Float box;
- Store/Recall/Clear Memory registers;
- Copy/Paste to/from the Clipboard;
- Click on the **Configure**, **Help** and **Exit**.
- Resize the window,

### See also:

[Keyboard display](#)  
[XCALC buttons](#)  
[Keyboard operation](#)  
[Configuring XCALC](#)



This sends you to the configuration screen.

You figured this one out yourself, right?

Yes, Exit. Stop. DestroyWindow().  
If active, the Help window will also be closed.

You click/drag one of the lower corners of the window to resize. Note that the resulting window will not be exactly the size shown by the "rubber bands": The XCALC window can only grow or shrink in steps. The resize corners are currently disabled when one of the "button" panes is at the bottom, so you can only do mouse resize when either the number display is on its own, or when you display the pane with the logo/error messages. (*Windows Help* will not survive this resize operation; see [bugs](#).)

## Memory Registers

XCALC is equipped with ten memory registers; each can hold one number in Decimal, Complex or Integer (Hex/Oct/Bin) format. The keys to control the memory are as follows (use the numeric keypad):

<b>Ctrl-<i>n</i></b>	Store the X register in memory cell <i>n</i> ( $0 \leq n \leq 9$ ).
<b>Alt-<i>n</i></b>	Recall memory cell <i>n</i> to the X register.
<b>Ctrl-Alt-<i>n</i></b>	Clear (zero) memory cell <i>n</i> .

As a reminder, you can optionally (see [Configuring XCALC](#)) show small digits on the bottom pane on the screen that represent the memory registers. If a register contains zero, the digit is shown in white, otherwise in red. When the memory display is turned on, you can also control the operations with the mouse by clicking on the small digits:

<b>LEFT click</b>	Store the X register in memory cell <i>n</i> ( $0 \leq n \leq 9$ ).
<b>RIGHT click</b>	Recall memory cell <i>n</i> to the X register.
<b>LEFT double click</b>	Clear (zero) memory cell <i>n</i> .

You can also start (see [Configuring XCALC](#)) a separate, movable popup window which shows the contents of the various memory registers. You can also specify that memory registers are saved between sessions. The registers are saved in binary format in the file XCALC.DAT in your *Windows* directory.

## Memory Popup Window

If you select to show the separate popup window for the memory registers, you add another option to the store/recall processes; the small panes that appear with the memory contents can be clicked for store/recall as follows:

<b>LEFT double-click</b>	Store the X register in the selected memory cell
<b>RIGHT double-click</b>	Recall the selected memory cell to the X register.

The popup window does not have any caption, but to move the window you can click/drag anywhere within the popup.

*Note:* The keyboard shortcut keys **Ctrl-Z** or **Ctrl-N** will quickly toggle the display of the popup window.

**See also:**

[Configuring XCALC](#)

## Configuring XCALC

In the Configure Dialogue Box you can make the following settings:

### General Options

*Sound (default On):*

Whether or not to get a beep when you make a maths error.

*Underline (default On):*

Whether or not the *Alt*-hotkey for the various buttons should be underlined.

*Icon Start (default Off):*

This lets you force XCALC to start in an iconised state.

*Keep In Front (default Off):*

This lets you force XCALC to stay a topmost window (always visible).

*Save Registers (default Off):*

Memory registers can be saved in a file between sessions.

### Memory

*Show Indicators (default Off):*

This lets you turn on or off the memory display digits (0-9) in the bottom pane of the display.

*Popup Window (default Off):*

This lets you turn on or off the popup memory display that shows the contents of the memory registers.

*Popup 1 or 2 Columns (default 2):*

This changes the layout of the popup window.

### Display

*Caption (default On):*

Show or hide the caption/system menu. With this *off*, the program takes on a somewhat non-standard behaviour.

*Keyboard Display (default On):*

Turns on/off the display of the function keys.

*Radix Buttons (default On):*

Turns on/off the Decimal/Complex/Hex/Octal/Binary buttons.

*Disp/Trig Buttons (default On):*

Turns on/off the Fix/Eng/Sci/Deg/Rad buttons.

*Fix/Pushbuttons (default On):*

Turns on/off the fixed point settings and the 3 pushbuttons (Configure, KeyHelp, Exit).

*Logo/Error Window (default On):*

Turns on/off the bottom pane with the memory indicators, error messages and logo.

### Other Options

*Stack Display (default 4):*

This sets the number of stack registers that are shown on the display, as well as the actual size of the stack. The value must be between 1 and 8 (included). Note that the actual stack never becomes smaller than 4, although you may choose to display only 1, 2 or 3 registers.

*Size (default 100%):*

This resizes the window of XCALC as a percentage of the default size: the default size of XCALC depends on the Display Bitmap in use (see below). The default setting of 100% and DIGITS1 is too big for a VGA display. You will not be allowed to specify a size that would be too small for a *Windows* window (as specified by a system parameter).

*Display Bitmap:*

Everything in XCALC is sized around the display bitmaps. There are currently 6 bitmaps you can select, with the respective sizes (pixels) per digit:

**DIGITS1** (the default, 19x29 pixels),  
**DIGITS2** (small and ugly, 8x10 pixels),  
**DIGITS3** (*Courier*, 15x21 pixels),  
**DIGITS4** (medium ugly, 12x18 pixels),  
**DIGITS5** (small Zapf Chancery, 11x15 pixels), and  
**DIGITS6** (larger **Zapf Chancery**, 18x25 pixels).

Experiment to get the display and size that suits you best. If you make any changes to the XCALC main display (such as adding/deleting a pane, changing the Size or the Bitmap) and press OK, XCALC will momentarily disappear from the screen while it is rebuilding the screen windows.



This program lets you hide the Window Caption to minimise the use of screen real-estate. If you do this, you lose the normal 'move-the-window-with-the-mouse' procedure, as well as the minimise button, the system menu and the caption active/inactive state display. Most of these functions are retained, however, in a somewhat non-standard way (not the system menu and the minimise button):

- You can move the window by click/drag anywhere on the number display;
- A little blue circle in the top left corner shows if the window is active;
- You can click this circle: *left double-click* = configure, *right double-click* = exit program.

## Who wrote this program

The XCALC program is written by

Bernt Ribbum  
Institute for Electro/Data  
Rogaland University Centre  
P.O. Box 2557 Ullandhaug  
4004 Stavanger  
Norway

Email: [bernt@hsr.no](mailto:bernt@hsr.no)

(If you have used earlier versions of XCALC, you may have noticed I've moved from Australia where I wrote versions up to 1.1.)

This is one of the very first programs I have ever written for MS *Windows*, and I have learned a lot during the development period. Thanks to Borland C++ (Versions 2.0 and 3.1) writing the program has been fun, and I am happy to share it with anyone who will use it (well, not the source code...).

So don't send me any money, send email instead, and tell me that you like the program! I would appreciate any constructive feedback (or shout me a beer the next time you visit Stavanger).

### Version History:

Sep 1992: Version 1.0  
Nov 1992: Version 1.01 (added clipboard functions)  
Nov 1992: Version 1.02 (separate VGA/SuperVGA dialogue boxes)  
Dec 1992: Version 1.1 (fixed overflow error, added keep in front, display only selected elements on screen, configure on system menu, memory popup window, memory save)  
Jan 1993: Version 1.1 rel. 2 (removed BWCC.LIB from XCALC\_A)  
Aug 1993: Version 1.2 (added keyboard display, greyed out inactive buttons, mouse sizing, more digits on memory popup, added percent and remainder functions, fixed a few bugs)

### See also

[Known Bugs](#)

## Bugs

The bug with overflow in the four basic maths functions from Version 1.0 has been corrected. The same is true with a silly error from Version 1.1 that made 500 factorial crash the program. Also, some off-the-limit experiments like  $10^{(1e308)}$ , or trying the zero'th root of a non zero number (the  $x^{\sqrt{y}}$  function), had the same unfortunate effect. If you now get a Maths Exception error, please tell me.

There are certain operations that, although mathematically sound, will not be allowed. One such is to try raising zero to any power. (The correct answer here should be zero if *power* is positive.)

The system menu disappears altogether if you choose to turn the caption *off*.

If *Windows Help* is active when you change the display layout (changing the size, for example), the help program will terminate.

There is also a minor bug in the program, that can occur when you turn the caption OFF and move the display with the mouse, or if you move the memory popup window, or if you resize XCALC with the mouse. Normally, when windows are moved, *Windows* will freeze all other screen updates. XCALC will not do this, so if something is happening on the screen while the windows are being moved, you may experience some distortion in the display of the underlying applications.

Apart from this, I'm happily unaware of anything.

## **A Maths Exception has occurred in XCALC**

This error happens if you generate an overflow or a singularity that is not trapped by another routine. The author honestly believes that such an error should never occur, but if it does and you see this, please contact me ([the author](#)) and report what happened.

Your only option now is to click OK, and the XCALC instance will be closed. Note that the program will exit in a normal manner, so all the settings (such as mode, position etc.) will be updated as usual.

