Cabamap

CAlculator **BA**sed **MA**thematics **P**latform
Version 1.02

Programmed By:
Michael Vincent
me@michaelv.org

Please send comments, bug reports, or suggestions to the above e-mail address.

Introduction

Cabamap is a flash application for the TI-83 Plus and TI-83 Plus Silver Edition which performs calculations involving integers, with arbitrary precision. This means that it will calculate the exact value of such operations such as 100 factorial, or 353290719*6235931419-358765.

Installation

Use the TI Connect or TI-GRAPHLINK software to send Cabamap.8xk to your calculator. If you are unsure of how to do this, check the documentation for MirageOS, also on the Detached Solutions website. The process to send an app is the same.

The First Run

Press the APPS key to open the Apps menu. Scroll to Cabamap and press enter.

The first time you run Cabamap, you will see the splash screen. Press any key to continue.

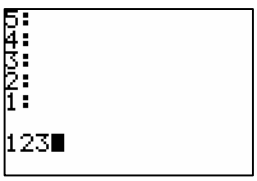
RPN

Cabamap is unlike traditional calculators in that it uses RPN for calculations. RPN, short for Reverse Polish Notation, is a postfix method of entering numbers where you first push numbers onto a stack, and then enter the operations to be performed on those numbers. If you need a tutorial on RPN, visit http://www.hpmuseum.org/rpn.htm

Stack:

The basic layout of the stack is shown below: The top five values on the stack are shown, although you may have up to 100 values on the stack. Typically four is the most that will be used in practical applications.

To enter a number on the stack, begin typing using digits 0-9. The number will appear in the working space at the bottom of the screen. The Del and left arrow keys will backspace. When you are finished, press Enter and the number will be “pushed” onto the stack. Think of the stack as a pile of note cards. By entering a number, you are writing it on a note card and dropping it on the top of the pile.



If you wish to duplicate the number, press the Enter key when there is already a number on the stack and you are not currently

entering a number. Then, the function of the Enter key is to duplicate the top value on the stack.

If you are entering a number and press Clear, the number will be erased. Pressing Clear while not editing a number will shift the stack up by discarding the top value (in position 1). Similarly pressing Del while not editing a number will also discard the top stack value.

Pressing the Up or Down arrow keys while not editing a number will rotate the stack in the desired direction. Pressing the graph variable (X,T,O,n) key will swap the top two stack values with each other.

To negate the top value on the stack, press the negative sign key (located to the left of Enter on the keyboard).

If you have more than five values on the stack, the colon in "5:" will change to an up arrow to indicate this.

Basic Calculations

Operations can be classified into two types, those that work on two numbers (dyadic) and those that work with one (monadic). The factorial function (e.g. 7! = 5040), only needs one number as input. On the other hand, addition or multiplication operates with two numbers.

Operations with two numbers will use positions 1 and 2 of the stack, with position 2 being the more important number.

Operations with one number will use position 1.

All operations return their result on the top of the stack in position 1.

You can interrupt a calculation at any time by pressing the ON key.

Keyboard Reference:

Numeric Editing Mode (entered by pressing 0-9):

- The number keys 0-9 enter a number.
- Enter will push the number onto the stack.
- The Left arrow/Del key will backspace.
- Clear will erase the number being entered.
- If you press an operation key, the number will be pushed onto the stack, and then the operator executed.
- 2nd+Quit will exit the app.

Stack Mode (when not in Numeric Editing Mode)

- Enter will duplicate the top stack value
- Up and Down rotate the stack
- X,T,θ,*n* (graph variable key) swaps the top two stack values
- Del and Clear will discard the top stack value
- 2nd+EE will display the top stack value in scientific notation (press any key to return to normal display).
- The WINDOW key toggles between defaulting to regular or scientific notation for displaying numbers.
- If the top stack value will not fit on the screen, the Left/Right arrow keys will scroll it. Left or right arrows surrounding the number will indicate scrolling ability. Hold the ON key while scrolling and it will scroll in "turbo mode."
- Pressing MATH will access the extra functions menu (discussed later).
- 2nd+Quit will exit the app.
- The STO key will exit the app, but also export the top stack value into the Ans variable in string format.

Function Reference:

Notation: S1 is the top stack value, the number in position 1. S2 is the number in position 2 on the stack.

Regular Functions (on the keyboard):

+ Addition. S1 & S2 are removed. S1+S2 is pushed into S1.
- Subtraction. S1 & S2 are removed. S2-S1 is pushed into S1.
X Multiplication. S1 & S2 are removed. S1*S2 is pushed into S1.
/ Division. S1 & S2 are removed. S2/S1 is calculated, the remainder discarded, and the quotient pushed into S1.
- Negation sign. The value of S1 is negated.
^ Exponentiation. S1 & S2 are removed. S2^S1 is pushed into S1.
x^2 Square. S1 is squared.
10^x Anti-logarithm. 10^S1 is stored in S1.
v Square root. The integer square root of S1 is calculated.
LOG Logarithm. Log base-S1 of S2 is taken and stored in S1.

Extended functions (press the MATH key to access the menu).

v Square root. The integer square root of S1 is calculated.
mod Modulo arithmetic. S2 mod S1 is calculated.
isPrime – See next section
gcd This calculates the greatest common divisor of S1 and S2.
lcm This calculates the least common multiple of S1 and S2.
logb This finds the integer logarithm of S2 with base S1.
! Factorial function, calculates S1!

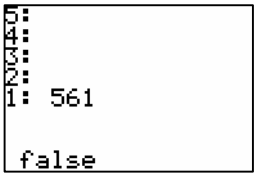
nPr Calculates the permutations of S2 nPr S1
nCr Calculates the permutation of S2 nPr S1

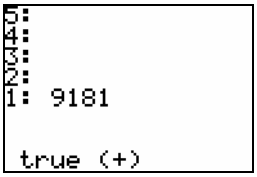
isPrime

This function is a primality test on the top stack value. It will return whether the number is prime or not. When run, it will display a number in parentheses on the last line of the screen – this number ascends from 0-11, informing the user of which of the 12 primality tests has just been completed.

After completing the tests (or interrupted by pressing ON), it will display true or false. If after completion the display shows false, the number is **definitely** not prime. If it is true, a confidence level will be shown in parentheses. For numbers smaller than 341,550,071,728,321, there will be a plus sign in parentheses. This indicates that the number is definitely prime.

For numbers larger than that, the algorithm will only be able to return that the number is probably prime. There remains an extremely small chance that the number is not prime. There will be a number shown in the parentheses, which is the confidence level. The higher the number, the more confident the calculator is that the number is prime. This allows you to press ON to break a prime test if it is taking a long time, and still have results.





If a number is listed as true (12), it has passed all 12 facets that the program tests via the Rabin-Miller algorithm. This means it is a 2,3,5,7,11,13,17,19,23,29,31,and 37-SPRP number.

Errors

There are a few errors that can occur. Error messages will be displayed in the last line of the screen. When shown, press Enter to clear the error and continue.

ERR: ARGUMENTS – You attempted to perform an operation which requires a certain number of values on the stack, and there aren't enough. For example, you will encounter this when trying to multiply with only one number on the stack.

ERR: MEMORY – The calculator is out of memory to complete the operation. You may want to try removing unneeded stack values if possible.

ERR: BREAK – You interrupted a calculation by pressing the ON key.

ERR: PARSE – This technically means that the calculator encountered invalid characters when attempting to parse an entered number. This should never occur though.

ERR: DIV BY 0 – You attempted to divide a number by zero.

ERR: DOMAIN – One or more inputs to an operator were not in the domain of the function. For example, you cannot take the factorial of a negative number.

ERR: STACK COUNT – There are too many values on the stack. You will have to remove a value before adding any more numbers.

Credits

I would like to thank Kirk Meyer for helping me decide which of the many possible algorithms for each function would best work in this project, the beta testing team for locating bugs, and Jason Malinowski for his last-minute assistance.

Version History

1.02 Released August 25, 2006
- Fixed a bug in the swapping routine

1.01 Released August 28, 2004
- Fixed a bug in the addition routine
- Fixed a stack exporting bug

1.0 Released February 21, 2004
- Limited the stack to 50 values to prevent overflows
- Added external interfacing ability

0.09 Released January 19, 2004
- Fixed bug in division routine when returning negative quotients in some cases
- Exporting number bug fixed
- Fixed an unusual bug in the square root routine
- Cabamap now properly strips away starting zeros (i.e. user enters 049).

0.08 Initial beta release