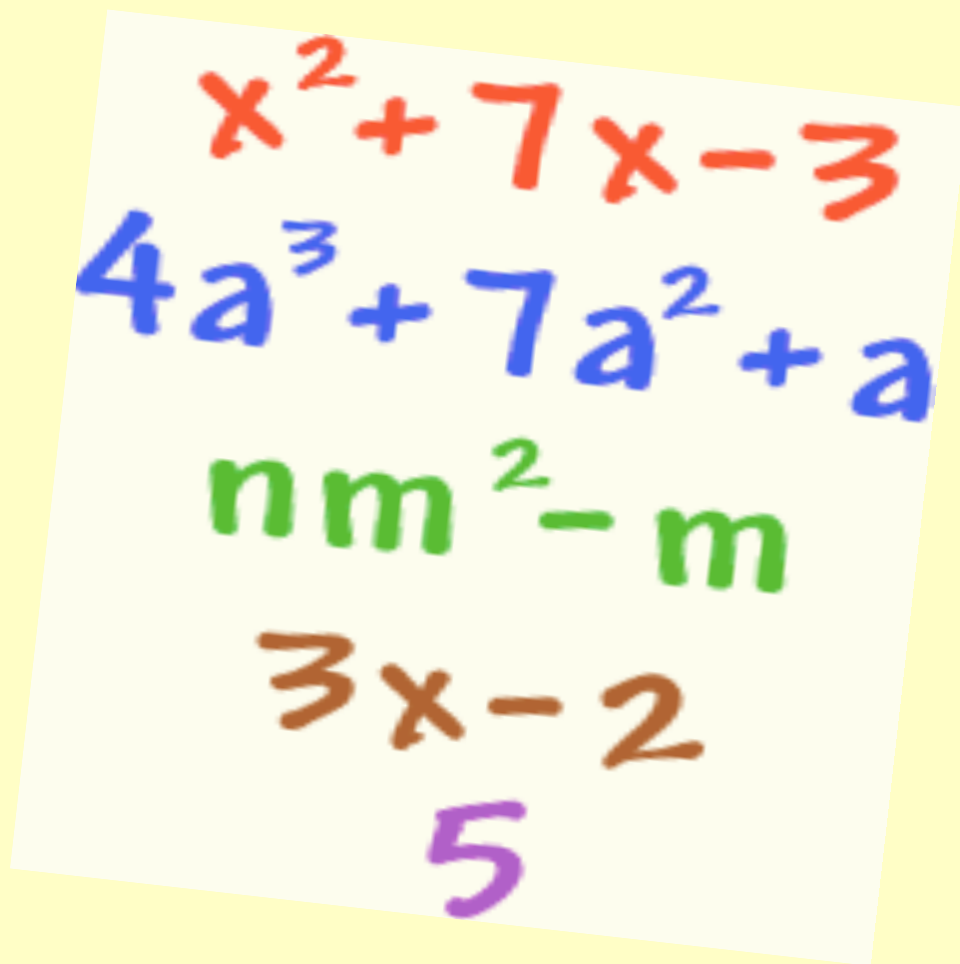


# Polynomials

## *All In One*



***Version 2.1***

By Anthony Cagliano

# Introduction

Thank you for downloading *Polynomials All in One*, a comprehensive polynomial math application developed by Anthony Cagliano, Internet alias 'ACagliano'. This is the third version and second remake of this tool. It is a stand-alone executable that will render the output of polynomial operations. Like its predecessor, this tool lets you input the polynomials as strings, not as lists the way many other similar tools require. However, unlike my last version, this one performs a series of polynomial operations via a prepended queue string (see the Usage section). Clocking in at just over 5kb, this program is a handy alternative to the bulky Flash Applications like Symbolic that provide the same features.

*This program is dedicated to one of the most amazing and inspirational friends someone could have. Not only does your friendship inspire me to be better in every way, but also something we talked about one day legit gave me the idea that brought this project to fruition. So thanks on two counts! :p*

## Download and Installation

If you are reading this, clearly you don't need help in downloading this program. However, if you have managed to breach the space-time continuum somehow and are able to read this and have no idea where to get the program, it may be downloaded from any of the following three places:

- <http://clrhqme.org/products/>
- <http://www.cemetech.net/programs/>
- <http://www.ticalc.org/pub/83plus/basic/math/>

Once you get the program, you will need to send it to your calculator using your favorite flavor of calculator connectivity software and a USB A to mini B connectivity cable. With the calculator plugged into the computer, launch the connectivity software and send POLYAIO2.8xp to your calculator in the manner indicated by your software.

## Usage

Using this program is fairly straightforward. Simply select it from the [Prgm] menu, and press [Enter] from the home screen. You will be presented with a prompt. See the image to the right.

```
Input Polynomial
or /h for help
or /q to quit
*****
█
```

This program lets you create a queue for processing the chain of polynomials to operate on. This queue consists of a two part input string, separated by a colon (:). The first part is the operator list. The second part is the operand list, with each operator enclosed in parenthesis. The table below shows what operators are used for what. Please note that for the *show work* option, the W must be the **first** character in the operator part of the string.

Operator	Function
+	Addition
-	Subtraction
*	Multiplication
/	Division
D or d	Derivative
I or i	Integral
Z or z	Zeroes
W or w	Show work

Three images are enclosed below, each giving an example of a different operation queue. In the one on the left, the second operand will be added to the first and then the third will be added to the result of that. In the middle one, the second operand will be multiplied by the first and then the third will be subtracted from that. In the far right example, operands one and two will be multiplied and then the derivative will be taken twice.

```
Input Polynomial
or /h for help
or /q to quit
*****
++:(X+1)(X+1)(X+1)█
```

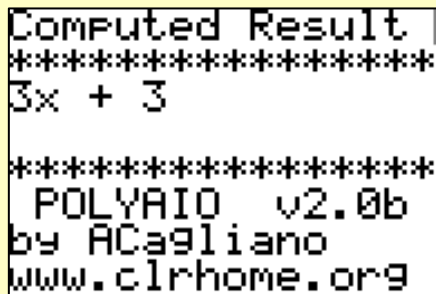
```
Input Polynomial
or /h for help
or /q to quit
*****
*-:(X^2-9)(X-5)(X+12)█
```

```
Input Polynomial
or /h for help
or /q to quit
*****
*DD:(X^2+2X+1)(X+1)█
```

Notice how, in the last example, the derivatives do not have operands to place on queue. This is because derivative operates only on the input and requires no secondary operand. Would there have been additional operators after the derivatives, the program would have resumed computing them after the derivatives, so long as additional operands were also present. The same is true for integrals and zeroes. The zeroes would function slightly different, displaying the zeroes of the result at that point, and then continuing with the queue if present.

This program has the ability to show work, defined as showing the result of the current operation. To activate this option, a “W” or “w” must be used as the **first** character in the operator string. It will show work on each iteration except the last.

The result will be returned after the entire queue has been



```
Computed Result
*****
3x + 3
*****
POLYAIO v2.0b
by ACagliano
www.clrhome.org
```

processed, in the manner indicated in the screenshot to the left, which corresponds to the far left image above. The result will be clipped where necessary and scrollable.

After pressing [Enter] you will be returned to the prompt screen, where you will be able to enter a new string or exit the program. To exit, simply type “/Q” or “/q”. To bring up the

help pages, type “/H” or “/h”.

## Algorithms

- String Conversion: Each polynomial operand is split into its coefficient and exponent mathematically for optimal speed. First, X is set to 1 and then the entire string is evaluated. 1 to any power evaluates to one, leaving the coefficient. Next, X is set to 2 and then the entire string is reevaluated and divided by the natural log of 2. Finally, once this process has been repeated for the entire polynomial, the coefficient list is sorted with respect to the exponent list, which is subsequently discarded.

- Addition/Subtraction: Nothing to this really. Lists are created, padded to equal length, and then the lists are either added or subtracted.
- Multiplication/Division: In exactly the same way as you would do these on paper, long multiplication and division are performed, complete with list shifting and all that jazz to make our job easier.
- Derivative/Integral: Chain and reverse chain methods are used for these. For the derivative, for each term, the coefficient and exponent are multiplied to give the new coefficient and the exponent is reduced by one. For the integral, for each term, the coefficient is divided by the exponent plus one, and the exponent is raised by one.
- Zeroes: Zeroes finding is done using a modified rational zeroes test. All factors of the coefficients of the first and last terms are divided and tested in the equation. All possible values that equate to zero are returned.

## Legal Notices

This program is released as freeware and with it comes full right to distribute and/or modify it to your heart's content. However, we strongly discourage you from distributing modified versions of this software because it will make it harder for us to troubleshoot any bugs that may arise.

By using this software, you release and save harmless ClrHome Productions and any of its affiliates from any liability for damages associated with its use. Such damage includes but is not limited to: (1) corruption or loss of data on your device, (2) crashing of the calculator, or "RAM clears", (3) corruption of the Flash memory or the operating system, or (4) in extreme cases, the permanent rendering of your device unusable, an event termed "bricking". However, please note that while we are required to cover worst-case scenarios, TI Basic programs are all but harmless.

We are also not responsible for any damage to your math grade that may result from improperly using this program or not following the instructions indicated herein. If you do not understand the instructions given in-program, come back here and consult this document. If you don't understand that, send me an email or contact me some other way listed below.

## Version History

### Polynomial Arithmetic Utility:

The fledgling version of this program, featuring input by lists and result by list. Broke on pretty much everything.

### Polynomials All in One:

A rewrite to the previous program, with faster and better algorithms. Input was to strings, which were then converted into lists. Each polynomial was entered separately and only one operation could be done at a time. Had sporadic bugs.

### Polynomials All in One v2:

An upgrade to version 1 of this program. Algorithms were streamlined. Specially formatted string allows for chaining operations. Improved interface.

- 2.0 b1 – First beta release of this software. Had only addition, subtraction, multiplication, and division working. Division by 1 term caused DOMAIN error.
- 2.0 b2 – Bugs removed from derivative and integral routines, memory recovery between iterations improved. New zeroes finding algorithm introduced.
- 2.0 RC1 – Division bug resolved. Error catching for divide by zero and negative exponents added. Trailing parenthesis removed where possible.
- 2.1 – Ability to show work added via a prepended function character (see Usage).

## Contact

Have a question? Comment? Complaint? Feature request? Bug report? Shoot me an email at [ac@clrhome.org](mailto:ac@clrhome.org). You can also find me on Facebook at <http://facebook.com/acagliano/> and on Twitter at <http://twitter.com/ACagliano/>.