

---

# CHAPTER 5

## NUMBERS AND CALCULATIONS

---

- Numbers and basic operators
  - Performing calculations
  - Using variables
  - Immediate mode
  - Numeric functions
  - Random numbers and other functions
- 

retroCombs



<https://www.stevencombs.com/plus4>

---

NUMBERS  
AND  
BASIC  
OPERATORS

You can use your Plus/4 like a simple calculator. Besides the standard + and - signs, your Plus/4 uses the \* sign for multiplication and the / sign for division and fractions. (Computers use the \* sign instead of an X for multiplication because a computer can't tell the difference between the letter X and the mathematical symbol  $\times$ .) You can use these operators and numbers in direct mode (no line numbers) or in a program. Neither numbers or operators should be in quotes for your Plus/4 to perform mathematical operations.

BASIC MATHEMATICAL OPERATORS			BASIC RELATIONAL OPERATORS	
Addition	+		Greater than	>
Subtraction	-		Less than	<
Division and fractions	/		Equals	=
Multiplication	*		Greater than or equal	>=
Exponentiation	↑		Less than or equal	<=
			Not equal to	<> or ><


NOTE: Your Plus/4 doesn't accept commas as part of a number. For example, you must type 109401 instead of 109,401. If you put a comma in a number, your Plus/4 thinks you mean two numbers (separated by the comma), so your Plus/4 would read 109 and 401 instead of 109401.

FRACTIONS  
AND  
DECIMALS

You can write a fraction like this: .5  
or like this: 1/2

Your Plus/4 is actually performing the division

If you put a fraction in a PRINT statement, your answer is always returned as a decimal or whole number. For example:

PRINT 139/493 + 5        
5.28194726

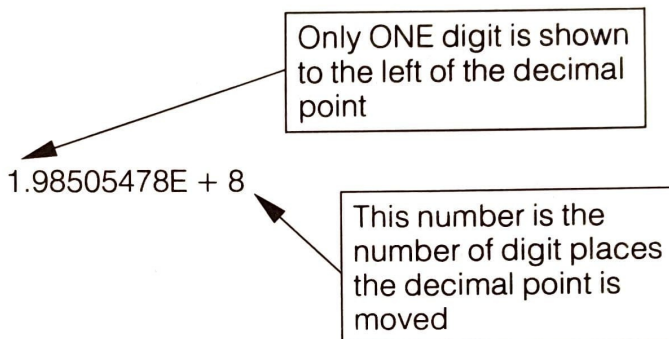
Here's an example that uses pi (3.14159256...), which represents the ratio of the circumference of a circle to its diameter. Use this value by just pressing the  $\pi$  key:

PRINT  $\pi$ /374        
8.39998036E-03

## SCIENTIFIC NOTATION

What did your Plus/4 mean by the E-03 part of the above answer? Your Plus/4 displays decimal numbers in the range –999,999,999 to 999,999,999 in standard numerals. Numbers beyond this range (with more than nine digits) are automatically displayed in scientific notation. You can enter numbers in yourself in this form and your Plus/4 will read them with no trouble (certainly less trouble than you had converting them). Scientific notation is often useful, since this special notation lets your Plus/4 display large numbers in fewer digits.

Here is how the number 198,505,478 would be written in scientific notation:



For a number less than one with several decimal places, the second number would be a – instead of a +, indicating that the decimal point is moved to the right.  
For example:

$$.0003359 = 3.359E - 4$$

Other examples:

$$20 = 2E + 1$$

the decimal point is moved 1 digit left

$$105000 = 1.05E + 5$$

the decimal point is moved 5 digits left

$$.0666 = 6.66E - 2$$

the decimal point is moved 2 digits right

## PERFORMING CALCULATIONS

To perform a calculation, type PRINT and then the math problem. Remember not to put the problem in quotes. Type this program:

NEW

10 PRINT 1+2, 2-1

20 PRINT 2\*2, 4/2

RUN

3        1  
4        2

use the slash on the ? key



For the first time, PRINT didn't print exactly what you typed in the statement. Instead, your Plus/4 solved the calculations and PRINTed the answers. All you have to do to use PRINT to calculate is omit the quotation marks. Now try this:

```
NEW
10 PRINT "2001/2010"
20 PRINT 2*3
RUN
2001/2010
```

↑ 6

one space is left here  
for the answer's sign

Since the calculation in line 10 is in quotes, your Plus/4 just PRINTs the problem as if it were regular text: exactly as it appears between the quotation marks. The problem isn't solved, and no space is left for the sign of the number.

Now move the cursor back to line 10 and change the line to this:

```
10 PRINT "2*3+1="";2*3+1
```

don't forget the semicolon

```
RUN
```

```
2*3+1 = 7
```

↓ this space is left for the  
answer's sign

```
6 ←
```

the answer for line 20  
stays the same

If you want to both PRINT the problem AND solve it you have to type it twice: once in quotes and once out of quotes, like this:

```
10 PRINT "2 + 2 = "; 2 + 2
```

## IMMEDIATE MODE

You can put any calculation in a program, or get an immediate answer by typing PRINT and the problem without a line number and pressing RETURN, like this:

```
PRINT 3-6
-3
PRINT 24/(6+2)
3
```

With numbers as well as with commands and text, when you don't have a line number before a BASIC statement, you




---

don't have to type RUN to tell the computer to follow the instruction; it's in IMMEDIATE, or DIRECT MODE. Having a line number means the statement is part of a BASIC program; it's in PROGRAM MODE. Either way is acceptable.


You can also include both a text statement in quotation marks and a mathematical problem to be solved in a single PRINT statement in immediate mode.

**PRINT "2 TO THE 3RD POWER  
EQUALS";2↑3**



this arrow stands for  
exponentiation; get  
it by typing SHIFT  
and 0

**2 TO THE 3RD POWER EQUALS 8**



the message is  
PRINTed and then  
the problem's solution is  
PRINTed

## **ORDER OF CALCULATION**

The second example in the last section shows that you can perform more than one calculation in one line. Try typing this:

**PRINT 200+ 50/5**

Is the answer what you expected? Try this:

**PRINT (200+ 50)/5**

Your Plus/4 always performs calculations in a certain order. Problems are solved from left to right; within that general rule, some types of calculations are solved first. The order which your Plus/4 evaluates expressions is called the order of precedence of operators.

FIRST: Your Plus/4 checks for negative numbers (not subtraction, just negative numbers).

SECOND: Your Plus/4 solves any exponents.

THIRD: Your Plus/4 solves all multiplications and divisions, from left to right.

FOURTH: Your Plus/4 solves additions and subtractions, from left to right.

NOTE: Your Plus/4 always solves any portion of the problem surrounded by parentheses first. You can even put parentheses within parentheses:  $36 * (12 + (A / 3))$ . The contents of the innermost parentheses are solved first.

Sometimes it's a good idea to put negative numbers in parentheses for clarity. For example, if you want to multiply 45 by  $-5$ , type it like this:  $45 * (-5)$ . Your Plus/4 can understand with or without parentheses.

## USING VARIABLES

The example  $36 * (12 + (A / 3))$  shows one of the most powerful features of a computer. When we used a letter instead of a number in a mathematical problem, we used a VARIABLE. A variable represents a value:

```
10 A = 3
20 PRINT "TOTAL: "; A * 4
```

If you RUN this program, the screen result is:  
TOTAL: 12

There are three types of variables you can use:

TYPE	SYMBOL	DESCRIPTION	EXAMPLES	SAMPLE VALUES
Floating point		real (decimal) or whole numbers	X, AB, T4	23.5, 12, 1.3E+2
Integer	%	whole numbers	X%, AI%	15, 102, 3
Text string	\$	letters, numbers, and all other characters in quotes	X\$, MS\$	"TOTAL:", "DAY 1", "CBM"

Every time you want a variable to be an integer variable, the symbol for that variable would include the % sign. A variable that contains text MUST end with a \$ as part of the variable. If it doesn't have that symbol, your Plus/4 considers it a floating point number. A variable without either of the symbols (% or \$) is read as a floating point number (a "regular" number). Integer variables are a subset of floating point variables; they are numbers with no decimal places.

Always use the right variable type. If you try to do something like assign a word to an integer variable, your program won't work. This program shows you what variable can or can't be used in a given situation, and you can find out what happens when you try out different types of data:



---

```
10 REM THIS PROGRAM NEEDS NUMERIC DATA
```

```
20 PRINT "ENTER A NUMBER"
```

```
30 INPUT X%
```

this is the variable  
to be input

```
40 PRINT "NICE GOING, ACE!"
```

```
50 "I READ YOUR NUMBER AS"; X%
```

Try to enter these values and see what happens:

ONE FIFTH

.043

10

## NUMERIC FUNCTIONS

Included in your Plus/4 BASIC 3.5 language are numeric functions, which are like the advanced calculations found on most scientific calculators (such as sine, cosine, tangent, etc.).

Most of the functions can be used by typing the name of the function and the number to be operated by the formula in parentheses, like this:

**FUNCTION(X)**

For example, to find out the sine of a variable, you would type:

**PRINT SIN(X)**

with X as any number you want to input.

You could also include one of the functions in a program line, as the following example shows:

```
10 FOR X= 1 TO 5
```

```
20 PRINT "THE SQUARE ROOT OF"; X;"IS"; SQR(X)
```

```
30 NEXT X
```

There is a complete listing of the numeric functions in the BASIC 3.5 Encyclopedia. Some of the more complex functions are explained in the following paragraphs.

## RANDOM NUMBERS AND OTHER FUNCTIONS

Selecting a random number is like taking 10 pieces of paper, writing a number from 1 to 10 on each piece and putting the 10 pieces of paper into a hat and drawing one piece of paper. The number chosen is a RANDOM number. The number is put back into the hat, and another number is drawn. Each time a number is drawn, it is put back in the hat, keeping the pool of possible numbers the same. When a number is selected, there is no way of knowing what number is going to come up next, but you do know that the number will be between 1 and 10. This is the basis of RANDOM NUMBERS.



---

Random numbers are extremely useful in programming, providing the element of chance or (obviously) randomness. Random numbers usually have a range, meaning there's an upper limit and a lower limit to the numbers you can draw. In the hat example, the range of numbers is 1 to 10. The lower limit is "1" and the upper limit is "10", which means that any number from 1 to 10 can come up randomly each time a new number is selected.

Let's examine how your Plus/4 handles random numbers, and some of the things you can do with them. This program generates five completely random numbers:

```
10 FOR X= 1 TO 5:PRINT RND(X):NEXT X
```

These random numbers are all rather complex, with several places on the right side of the decimal point... but most uses for random numbers require whole numbers. You can make your numbers come out as whole numbers (without decimal places) by using the INTeger function, which cuts off all the digits on the right side of the decimal point. The following gives you a formula for generating random numbers in any range you want. You can use this formula almost anywhere you would use a variable or number in your program.

**INT(range\*RND(1))** lower limit

The INT command tells the computer to cut off any decimal places and only give you whole numbers like 1, 45, or 320, instead of numbers like 1.223, 45.6677, or 320.59. Whole numbers are easier to work with when using random numbers.

**Lower Limit** in the formula refers to the lowest number you want the computer to choose from.

**Range** is how many numbers are in the total group.

For example, if you want to choose a random number from 1 to 5, the lower limit is 1 and the range is 5. If you want to choose a random number from 15 to 20, the lower limit would be 15 and the range is 6, because you are choosing from a pool of 6 numbers. If you're choosing numbers from 2 to 100, the lower limit is 2 and the range is 99. Now let's try out a program:

```
10 PRINT INT(5* RND(1)) + 1
```


Type RUN and press **RETURN**. RUN the program a few times. Each time you run the program, you get a random number from 1 to 5. Now let's print 15 random numbers, with the lower limit 1 and the range 5... note that all 15 numbers chosen are selected at random from between 1 and 5:

---

```
10 FOR X= 1 TO 15
20 PRINT INT(5*RND(1)) + 1
30 NEXT X
```

sets loop for 15 times

selects RaNDom  
number

Type **RUN** and press .

An effective way to use this formula is to make it into a **user defined function**. User defined functions are extremely useful in mathematical calculations, and extremely easy to implement using your Plus/4. User defined functions allow you to program a formula, and then let your Plus/4 plug in values to be calculated. This can be used for many different purposes. Section 10 of the Encyclopedia section contains a listing of mathematical function derivatives which can be used to define functions.

Here is a statement utilizing the user defined function for generating random numbers:

```
10 DEF FNR(X)=INT(X*RND(1)) + 1
```

This gives us random numbers in the range from 1 to X. FNR is the name of the function defined by this statement.

EXAMPLE using a defined function:

```
10 DEF FNR(X)=INT(X*RND(1)) + 1
20 DO
30 COLOR 1, FNR(16), 5: REM PICK A COLOR FROM 1
  TO 16
40 PRINT "THE SEARCH GOES ON"
50 LOOP
```

Using the defined function saves memory space when you use the function more than once, and makes your programs easier to read and understand.