Chapter 2: Math, Angle, and Test Operations

This chapter describes math, angle, and relational operations that are available on the TI-82. The most commonly used functions are accessed from the keyboard; others are accessed through full-screen menus.

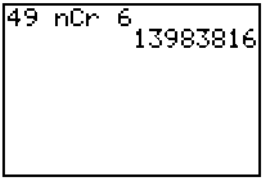
Chapter Contents

Getting Started: Lottery Chances

Getting Started is a fast-paced introduction. Read the chapter for details.

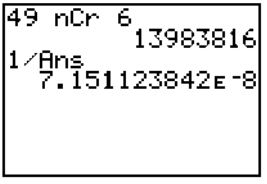
Suppose you want to enter a lottery where 6 numbers will be drawn out of 49. To win, you must pick all 6 numbers (in any order). What is the probability of winning if you buy one ticket? What is the probability of winning if you buy five tickets?

1. Determine the number of combinations possible. On the Home screen, press **49** to enter the total number of items. Press MATH ◀ to display the MATH PRB menu. Press **3** or ▼ ▼ ENTER to select **nCr**. Press **6** to enter the number of items selected.

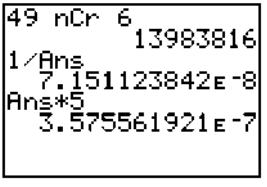


2. Press ENTER to evaluate the expression. This is the total number of possible combinations of 6 numbers drawn from a set of 49 numbers. With one ticket, you have one chance in 13,983,816 of winning.

3. To calculate the probability of winning with one ticket, press **1** ÷ 2nd [ANS] ENTER. The answer is expressed in scientific notation on the TI-82 because it is so small. The decimal equivalent is 0.00000007151123842.



4. To calculate the probability of winning with five tickets, press × 5 ENTER. Again, the answer is too small to display in fixed notation. The decimal equivalent is 0.0000003575561921.

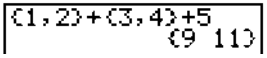


Keyboard Math Operations

The most commonly used math functions are on the keyboard.

Using Lists with Functions

Functions that are valid for lists return a list calculated on an element-by-element basis. If two lists are used in the same expression, they must be the same length.



+ (Add), – (Subtract), ∗ (Multiply), / (Divide)

+ (addition [+]), **–** (subtraction [−]), ∗ (multiplication [×]), and / (division [÷]) may be used with numbers, expressions, lists, or matrices (Chapter 10).

valueA+*valueB*, *valueA*–*valueB*, *valueA*∗*valueB*, *valueA*/*valueB*

Trig Functions

The trigonometric functions may be used with numbers, expressions, or lists. They are interpreted according to the current **Radian**/**Degree** MODE setting. For example, **sin 30** in **Radian** MODE returns **-.9880316241**; in **Degree** MODE it returns **.5**.

sin *value*, **cos** *value*, **tan** *value*

sin^{-1}, **cos^{-1}**, and **tan^{-1}** are the inverse trig functions (arcsine, arccosine, and arctangent).

sin^{-1} *value*, **cos^{-1}** *value*, **tan^{-1}** *value*

^ (Power), ^{2} (Square), √ (Square Root)

^ (power [^]), **^{2}** (squared [x^2]), and √ (square root [2nd] [√]) may be used with numbers, expressions, lists, or matrices (Chapter 10).

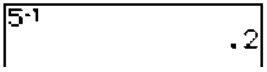
value^*power*, *value*^{2}, √*value*

Note: Raising a negative number to a noninteger power can result in a complex number, which returns an error.

$^{-1}$ (Inverse)

$^{-1}$ (inverse [x^{-1}]) may be used with numbers, expressions, lists, or matrices (Chapter 10).The multiplicative inverse is the equivalent of the reciprocal, 1/x.

value^{-1}



log, 10^, ln

log (logarithm LOG), **10^** (power of ten 2nd [10^x]), and **ln** (natural log 2nd [ln]) may be used with a number, expression, or list.

log *value*, **10^***power*, **ln** *value*

e^

e^ (exponential 2nd [e^x]) may be used with a number, expression, or list. **e^** returns the constant e raised to a power. **e^1** returns the value of the constant e.

e^*power*

```
e^5
      148.4131591
e^1
      2.718281828
```

- (Negation)

- (negation (-)) returns the negative of a number, expression, list, or matrix (Chapter 10). The narrow negation symbol (**-**) distinguishes negation from the subtraction or minus − (−).

-*value*

EOS rules (Chapter 1) determine when negation is evaluated. For example, **-A²** returns a negative number (squaring is evaluated before negation according to EOS rules). Use parentheses to square a negated number, **(-A)²**.

```
2→A:{-A²,(-A)²,-
2²,(-2)²}
      {-4 4 -4 4}
```

abs

abs (absolute value) returns the absolute value of a number, expression, list, or matrix (Chapter 10).

abs *value*

π (Pi)

Pi is stored as a constant in the TI-82. Press 2nd [π] to copy the symbol π to the cursor location. The number 3.1415926535898 is used internally in calculations.

MATH MATH Operations

To display the MATH MATH menu, press [MATH]. When you select an item from the menu, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.

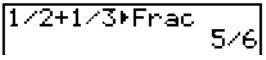
MATH MATH Menu

MATH NUM HYP PRB	
1: ▶Frac	Display answer as fraction
2: ▶Dec	Display answer as decimal
3: ^{3}	Cube
4: $\sqrt[3]{}$	Cube root
5: $\sqrt[x]{}$	n^{th} root
6: fMin(	Minimum of a function
7: fMax(	Maximum of a function
8: nDeriv(	Numerical derivative
9: fnInt(	Function integral
0: solve(	Solution (root) of a function

▶Frac

▶Frac (display as fraction) displays an answer as the rational equivalent. The answer may be a number, expression, list, or matrix. If it cannot be simplified or the denominator is more than three digits, the decimal equivalent is returned. **▶Frac** is valid only at the end of an expression.

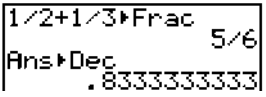
expression▶**Frac**



▶Dec

▶Dec (display as decimal) displays an answer in decimal form. **▶Dec** is valid only at the end of an expression.

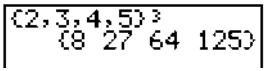
expression▶**Dec**



^{3} (Cube)

^{3} (cube, MATH MATH item 3) returns the cube of a number, expression, list, or square matrix (Chapter 10).

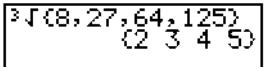
value^3



$\sqrt[3]{}$ (Cube Root)

$\sqrt[3]{}$ (cube root, MATH MATH item 4) returns the cube root of a number, expression, or list.

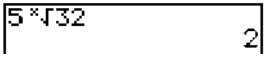
$\sqrt[3]{}$*value*



$\sqrt[x]{}$ (Root)

$\sqrt[x]{}$ (root, MATH MATH item 5) returns the n^{th} real root of a number, expression, or list.

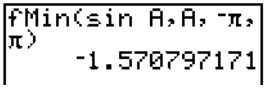
$n^{th}root\sqrt[x]{}value$



fMin(, fMax(

fMin((function minimum, MATH MATH item 6) and **fMax(** (function maximum, MATH MATH item 7) return the value at which the minimum or maximum value of *expression* with respect to *variable* occurs, between *lower* and *upper* values for *variable*. *lower* must be less than *upper*. **fMin(** and **fMax(** are not valid in *expression*. The accuracy is controlled by *tolerance* (optional; if not specified, 1E⁻5 is used). If there is no finite minimum or maximum in the interval, usually (depending on *expression*) an error occurs.

fMin(*expression*,*variable*,*lower*,*upper***)** or
fMin(*expression*,*variable*,*lower*,*upper*,*tolerance***)**



nDeriv(

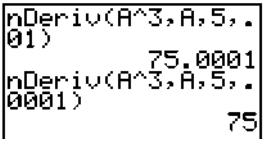
nDeriv((numerical derivative, MATH MATH item 8) returns an approximate derivative of *expression* with respect to *variable*, given the *value* at which to calculate the derivative, and ε (optional; if none is specified, 1E-3 is used).

nDeriv(*expression*,*variable*,*value***)** or
nDeriv(*expression*,*variable*,*value*,ε**)**

nDeriv(uses the symmetric difference quotient method, which approximates the numerical derivative value as the slope of the secant line through the points:

(*value*–ε, *expression*(*value*–ε)) and
(*value*+ε,*expression*(*value*+ε))

As ε gets smaller, the approximation usually gets more accurate.

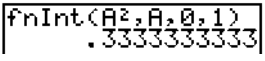


nDeriv(can be used once in *expression*. Because of the method, **nDeriv(** can return a false derivative value at a nondifferentiable point.

fnInt(

fnInt((function integral, MATH MATH item 9) returns the numerical integral (Gauss-Kronrod method) of *expression* with respect to *variable*, given *lower* limit, *upper* limit, and a *tolerance* (optional; if none is specified, 1E-5 is used).

fnInt(*expression*,*variable*,*lower*,*upper***)** or
fnInt(*expression*,*variable*,*lower*,*upper*,*tolerance***)**



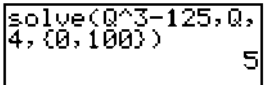
fnInt(is not valid in *expression*.

solve(

solve((MATH MATH item 0) returns a solution (root) of *expression* for *variable*, given an initial *guess*, a *lower* bound, and an *upper* bound within which a solution is sought (optional, if not specified, *lower*=-1E99 and *upper*=1E99).

solve(*expression*,*variable*,*guess***)** or
solve(*expression*,*variable*,*guess*,**{***lower*,*upper***})**

expression is assumed equal to zero. The value of *variable* in memory will not be updated. *guess* may be a value or a list of two values. Values must be stored to every variable in *expression*, except *variable*, before *expression* is evaluated. *lower* and *upper* are entered in list format.



Controlling the Solution for solve(

The TI-82 solves equations through an iterative process. To control that process, you should provide a close bound of the solution and at least one initial guess (which must be within the bounds). This will help to:

- Find a solution.
- Define which solution you want for equations with multiple solutions.
- Find the solution more quickly.

MATH NUM (Number) Operations

To display the MATH NUM menu, press MATH ▸. When you select an item from the menu, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.

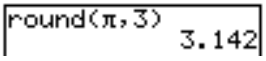
MATH NUM Menu

MATH NUM HYP PRB	
1: round(	Round
2: iPart	Integer part
3: fPart	Fractional part
4: int	Greatest integer
5: min(	Minimum value
6: max(	Maximum value

round(

round(returns a number, expression, list, or matrix rounded to *#decimals* (≤9). If *#decimals* is omitted, *value* is rounded to 10 digits.

round(*value*,*#decimals***)**
round(*value***)**



iPart

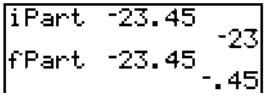
iPart (integer part) returns the integer part or parts of a number, expression, list, or matrix (Chapter 10).

iPart *value*

fPart

fPart (fractional part) returns the fractional part or parts of a number, expression, list, or matrix (Chapter 10).

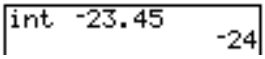
fPart *value*



int

int (greatest integer) returns the largest integer less than or equal to a number, expression, list, or matrix. The value is the same as **iPart** for nonnegative numbers and negative integers, but one integer less than **iPart** for negative noninteger numbers.

int *value*

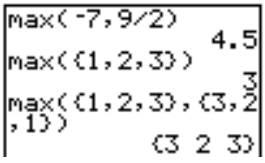


min(, max(

min((minimum value) returns the smaller of *valueA* or *valueB* or the smallest element in a list. If two lists are compared, it returns a list of the smaller of each pair of elements.

max((maximum value) returns the larger of *valueA* or *valueB* or the largest element in a list. If two lists are compared, it returns a list of the larger of each pair of elements.

min(*valueA*,*valueB***)** or **max(***valueA*,*valueB***)**
min(*list***)** or **max(***list***)**
min(*listA*,*listB***)** or **max(***listA*,*listB***)**



Note: The **min(** and **max(** functions on the MATH NUM menu are the same as the **min(** and **max(** functions on the LIST MATH menu.

MATH HYP (Hyperbolic) Operations

To display the MATH HYP menu, press MATH ▶ ▶. When you select an item from the menu, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.

MATH HYP Menu

MATH NUM HYP PRB	
1: sinh	Hyperbolic sine
2: cosh	Hyperbolic cosine
3: tanh	Hyperbolic tangent
4: $\sinh^{-1}$	Hyperbolic arcsine
5: $\cosh^{-1}$	Hyperbolic arccosine
6: $\tanh^{-1}$	Hyperbolic arctangent

sinh, cosh, tanh

sinh, **cosh**, and **tanh** are the hyperbolic functions. They are valid for lists.

sinh *value*

```
sinh .5
     .5210953055
```

$\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$

$\sinh^{-1}$, **$\cosh^{-1}$**, and **$\tanh^{-1}$** are the hyperbolic arcsine, hyperbolic arccosine, and hyperbolic arctangent functions, respectively. They are valid for lists.

$\sinh^{-1}$ *value*

```
sinh⁻¹ {0,1}
  {0 .8813735870}
```

MATH PRB (Probability) Operations

To display the MATH PRB menu, press [MATH] [◀]. When you select an item from the menu, the name is copied to the cursor location. Functions that are valid for lists return a list calculated on an element-by-element basis.

MATH PRB Menu

MATH NUM HYP PRB	
1:rand	Random number generator
2:nPr	Number of permutations
3:nCr	Number of combinations
4:!	Factorial

rand

rand (random number) generates and returns a random number greater than 0 and less than 1. A random number is generated from a seed value. To control a random number sequence, first store an integer seed value in **rand**. If you store **0** to **rand**, the TI-82 uses the factory-set seed value. When you reset the TI-82, **rand** is set to the factory seed.

```
0→rand:rand*3
      2.830792207
```

nPr

nPr (number of permutations) returns the number of permutations of *items* taken *number* at a time. *items* and *number* must be nonnegative integers.

items **nPr** *number*

nCr

nCr (number of combinations) returns the number of combinations of *items* taken *number* at a time. *items* and *number* must be nonnegative integers.

items **nCr** *number*

```
5 nPr 2
                20
5 nCr 2
                10
```

! (Factorial)

! (factorial) returns the factorial of a positive integer between 0 and 69.

*value***!**

```
6!
               720
```

ANGLE Operations

To display the ANGLE menu, press 2nd [ANGLE]. The ANGLE menu displays angle indicators and instructions. When you select an item from the menu, the name is copied to the cursor location. Angle entries are interpreted according to the Radian/Degree MODE setting.

ANGLE Menu

ANGLE	
1: °	Degree function
2: '	DMS entry notation
3: ʳ	Radian function
4: ▶DMS	Display as degree/minute/second
5: R▶Pr(	Returns **R**, given **X** and **Y**
6: R▶Pθ(	Returns θ, given **X** and **Y**
7: P▶Rx(	Returns **X**, given **R** and θ
8: P▶Ry(	Returns **Y**, given **R** and θ

Note: Do not enter DMS numbers as **54°32'30"** on the TI-82. **54°32'** is interpreted as implied multiplication of 54° and 32', and **"** is a quote mark used to enter text.

°(Degree)

°(degree) lets you designate *angle* as degree, regardless of the current angle MODE setting. *angle* may be a list.

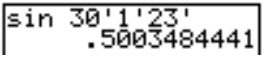
angle°

' (DMS Entry Notation)

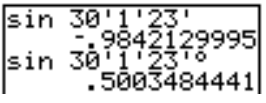
' (DMS entry notation) lets you enter degrees, minutes, and seconds in DMS format.

degrees'*minutes*'*seconds*'

For example, enter **30'1'23'** for 30 degrees, 1 minute, 23 seconds. Note that the MODE setting must be **Degree** (or you must use the **Degree** function) for the TI-82 to interpret the argument as degrees, minutes, and seconds.



Degree MODE



Radian MODE

ʳ (Radians)

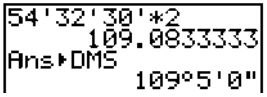
ʳ (radian) lets you designate *angle* as radian, regardless of the current angle MODE setting. *angle* may be a list.

*angle*ʳ

▶DMS

▶DMS (display as degree/minute/second) displays *answer* in degree, minute, second format. The MODE setting must be **Degree** for the TI-82 to interpret *answer* as degrees, minutes, and seconds. **▶DMS** is valid only at the end of a line.

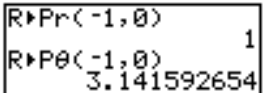
answer▶**DMS**



R▶Pr(, R▶Pθ(, P▶Rx(, P▶Ry(

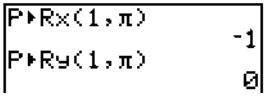
R▶Pr(converts rectangular to polar and returns **R**, and **R▶Pθ(** converts rectangular to polar and returns θ, given *X* and *Y* rectangular coordinate values.

R▶Pr(*X*,*Y***)**
R▶Pθ(*X*,*Y***)**



P▶Rx(converts polar to rectangular and returns **X**, and **P▶Ry(** converts polar to rectangular and returns **Y**, given *R* and *θ* polar coordinate values.

P▶Rx(*R*,*θ***)**
P▶Ry(*R*,*θ***)**



TEST TEST (Relational) Operations

To display the TEST TEST menu, press 2nd [TEST]. When you select from the menu, the name is copied to the cursor location. These functions are valid for lists; they return a list calculated on an element-by-element basis.

TEST TEST Menu

TEST LOGIC	True if:
1: =	Equal
2: ≠	Not equal to
3: >	Greater than
4: ≥	Greater than or equal to
5: <	Less than
6: ≤	Less than or equal to

=, ≠, >, ≥, <, ≤

Relational operators compare *valueA* and *valueB* and return **1** if the test is true or **0** if the test is false. *valueA* and *valueB* can be numbers, expressions, lists, or matrices (Chapter 10), but they must match in type and dimension. Relational operators are often used in programs to control program flow and in graphing to control the graph of a function over specific values.

valueA=*valueB*

```
{1,2,3}={3,2,1}
              {0 1 0}
{1,2,3}≠{3,2,1}
              {1 0 1}
{1,2,3}<{3,2,1}
              {1 0 0}
```

Using Tests

Relational operators are evaluated after mathematical functions according to EOS rules (Chapter 1).

- The expression **2+2=2+3** returns **0**. The TI-82 does the addition first because of EOS rules, and then it compares 4 to 5.
- The expression **2+(2=2)+3** returns **6**. The TI-82 first performs the relational test because it is in parentheses, and then it adds 2, 1, and 3.

TEST LOGIC (Boolean) Operations

To display the TEST LOGIC menu, press 2nd [TEST] ▸. When you select from the menu, the name is copied to the cursor location.

TEST LOGIC Menu

TEST LOGIC	True if:
1:and	Both values are nonzero (true)
2:or	At least one value is nonzero (true)
3:xor	Only one value is zero (false)
4:not	The value is zero (true)

Boolean Operators

Boolean operators are often used in programs to control program flow and in graphing to control the graph of a function over specific values. Values are interpreted as zero (false) or nonzero (true).

and, or, xor

and, **or**, and **xor** (exclusive or) return a value of **1** if a expression is true or **0** if the expression is false, according to the table below. *valueA* and *valueB* can be expressions.

valueA **and** *valueB*
valueA **or** *valueB*
valueA **xor** *valueB*

valueA	*valueB*		**and**	**or**	**xor**
≠0	≠0	returns	1	1	0
≠0	0	returns	0	1	1
0	≠0	returns	0	1	1
0	0	returns	0	0	0

not

not returns 1 if *value* (which can be an expression) is 0.

not *value*

Using Boolean Operations

Boolean logic is often used with relational tests. In a program, the following instructions store **4** into **C**:

```
PROGRAM:BOOLEAN
:2→A:3→B
:If A=2 and B=3
:Then:4→C
:Else:5→C
:End
```