# The usage of RealNumberCalculator

This is the manual of a scientific calculator library I published in Nuget with this title name.

Anyways, the reason to publish was only a try to fix trouble. At first this library did not work and I wondered if Nuget package form would work. But the trouble cause was only a version difference of '.NET' .

So I did not intend it to be used by the general public. But it has been downloaded from some people for any reason. I decided to hold it open and write this document because it might become the advertisement of my application.

This is a part of commercial application I have developed. I have not abandoned the rights relationship and I have not opened the source program. Also, all of it is developed by myself and I have not merged with any part of the open source program exposed in public.

Do not use this program for not-personal use or reverse engineering.

Program example C# of this file last is a scientific calculator available in Windows command prompt. It calculates formula string inputted in standard input, and outputs the result to standard output. Make a text file consisted of lined up formulas, and input "command < input file name > output file name". A file calculation results are added to each line, is made. UTF-8 code is used and previous "chcp 65001" execution is necessary. I have been using this library in my application by passing a formula string. It seems that the usage beyond this example is not normal.

The function and command list start from 3 page. Its ability is the level of fundamental scientific calculator+$\alpha$. However, it doesn't equip the kind of distribution, hypothesis testing.

As a characteristic of this calculator, it calculates with **decimal number 16-digits** through whole function. It doesn't include binary conversion error by that. In my calculation confirmation, its function returns more accurate result than double. It detects that the value exceeds 16-digits or not throughout the calculation. It can show you that the result is accurate or not. In the limitation to Addition/Subtraction/Multiplication/Division, when the result exceeds 16-digits, it can expand the result until 34-digits.

At division, it automatically converts the value to **fractions**, and holds and calculates by the shape. And it holds square root or circular constant in number and calculates with the form, to some extent. It is particular about the calculation precision in this way.

If seeing only number length, Windows calculator shows 32-digits to whole function results, but binary calculation. I am sure this spec best meets the function accuracy requirements.

It equips not a serious computer algebra system, but a limited imitation. Only to formula of variable X, it can simplify, do derivation, and do integration. (It can only do simple integration by parts, or integration by substitution. Serious CAS integration seems to equip with huge libraries.)

Don't hesitate to try it if you are a skillful adult who can use by yourself with this explanation only.
I have intention to fix bugs if you report to me. But I can make no assurance even if you may have got something damage from it by commercial use. It has more functions than I have used in my application and I have not tested all of them. The command list has no explanation, but I have a wish to make a more expanded document if my commercial scientific calculator sells a lot.

## The words and letters available in this calculator

(I added explanation I think necessary, only to peculiar to this calculator. Almost functions follow normal scientific calculator's.)

### Numeric input/result

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, . , −, e

~ : When number includes error

` : Recurring part start location in recurring fraction decimal

″,″,″ ″ :  Numeric inside delimiter (Punctuation)

_, / : Fraction delimiter

√ : Square root (Irrational number)

### Function input

sin, cos, tan, sin$^{-1}$, cos$^{-1}$, tan$^{-1}$, sinh, cosh, tanh, sinh$^{-1}$, cosh$^{-1}$, tanh$^{-1}$, sec, csc, cot, sec$^{-1}$, csc$^{-1}$, cot$^{-1}$

log$_{10}$, log$_e$, log$_2$, pow10^, powe^, pow2^, abs, int, ipart, fpart, $^3$√, prime : Functions in front of number

＋, —, ×, ÷, ＋%, —%, ×%, ÷%, int÷, mod, ^ , $^x$√, nCr, nPr, GCM, LCM, =, ≠, ＞, ≥, ＜, ≤

: Functions between numbers

^$^{-1}$, ^$^2$, ^$^3$, ！, % : Functions after number

% : Percent calculation execution

→Frac : Conversion to fraction

Frac( : Conversion to fraction (Appoint Denominator)

log(

round( : Rounding

### Formula input

(, )

; : Separator between numeric in ( ) [',' is separator in numeric]

if( : Condition branch

| : Multi-statement delimiter

### Constant

π, e

Φ : Golden Number

### Scientific Constant

@c0, @μ0, @ε0, @G0, @G, @h, @ħ, @e, @Φ0, @me, @mp, @mp/me, @α, @α−1, @R∞, @NA, @F, @R, @κ, @σ,
@eV, @u, @Z0, @μB, @a0, @μN, @Eh, @h/2me, @c1, @c2, @gn, @mn, @mμ, @Vm, @RK

Random number

RANDOM,R.DICE,R.COIN,R.INT

Random number function

rnd-minmax(

Memory

Ａ,Ｂ,Ｃ,Ｄ,Ｅ,Ｆ,Ｇ,Ｈ,Ｉ,Ｊ,Ｋ,Ｌ,Ｍ,Ｎ,Ｏ,Ｐ,Ｑ,Ｒ,Ｓ,Ｔ,Ｕ,Ｖ,Ｗ,Ｘ,Ｙ,Ｚ,θ

Val0,Val1,Val2,Val3,Val4,Val5,Val6,Val7,Val8,Val9

Storing number in memory

➡,M+,M-

Answer memory

ANS

Answer memory command

AnswerON,AnswerOFF : Storing result in Answer memory or not

SI Prefix input/result

da,h,k,M,G,T,P,E,Z,Y,d,c,m,μ,n,p,f,a,z,y : Length

$da^2,h^2,k^2,M^2,G^2,T^2,P^2,E^2,Z^2,Y^2,d^2,c^2,m^2,\mu^2,n^2,p^2,f^2,a^2,z^2,y^2$ : Area

$da^3,h^3,k^3,M^3,G^3,T^3,P^3,E^3,Z^3,Y^3,d^3,c^3,m^3,\mu^3,n^3,p^3,f^3,a^3,z^3,y^3$ : Volume

Ki,Mi,Gi,Ti,Pi,Ei,Zi,Yi : Binary Prefix

□ : No Prefix

SI Prefix command

PfxDim : Appoint Area/Volume

Result command

ImprFrac,MixedFrac,Decimal,Recurring

: Improper fraction,Mixed fraction,Fraction decimal,Recurring fraction decimal

Digit : Number digit length

Fraction decimal result command

Tab : Fraction decimal part length

RoundDown,RoundOff,RoundUp,RoundFloor,RoundCeil,RoundEven : The way of Rounding

Normal,Fix,Sci,Eng : The way of decimal display

Prefix : Result display with SI Prefix

Exact : High accuracy result display until 16 or 34-digits

PuncON,PuncON2,PuncON4,PuncOFF : Punctuation location

FracPuncON,FracPuncOFF : Punctuation in Fraction decimal part or not

DoubleON,DoubleOFF : When Addition/Subtraction/Multiplication/Division result exceeds 16 digits, automatically expand until 34 digits or not

ExpInLen,ExpOutLen : Exponential part is counted to number length, or not

EtcShow,EtcHide : '~' mark for error is shown in result or not

Fraction result command

FracInLen,FracOutLen : Fraction is shown only when within digit length, or not

Irrational number result command

Commonenomi,IndiDenomi : Reduce to a common denominator or not

Recurring fraction decimal result command

RecurParensis,RecurMark : Recurring part is in parentheses or not

SI Prefix result command

Prefix12ON,Prefix12OFF : Use exponent 1,2,-1,-2 Prefix in result, or not

Hexadecimal number input/result

A,B,C,D,E,F : Hexadecimal number significant digits are **15-digits**

p : In hexadecimal expression of binary exponential number(C-language printf), appoint exponent part

Positional notation number base prefix input/result

hex,duo,dec,oct,qui,qua,bin

Positional notation number conversion

→hex,→duo,→dec,→oct,→qui,→qua,→bin

Positional notation number result command

BaseON,BaseOFF : Show number base prefix or not

Complement,HexMinus : Negative number result is Complement or not

Fullway,Halfway : Cut result in digit length or not

MagnifyON,MagnifyOFF : Magnify digit count or not

Logic function

NOT,NEG

AND,OR,XOR,NAND,NOR,XNOR

Unit(Angle) input/result

°,′,″,rad,grade,cycle

Unit(Angle) conversion

→°,→′,→″,→rad,→grade,→cycle

Unit(Angle) result command

DegMinSec,Degree,Radian,Gradian,Cycle

Unit(Time) input/result

hour,min,s

AM,PM

Unit(Time) result command

Time12,Time24

Unit(Date) input/result

year,month,day

Unit(Date) result command

DateLong,DateShort,DateZero,DateSpace

DateYMD,DateMDY,DateDMY,Days360,Days365

Unit(Temperature) conversion

° F→° C,° C→° F,° F→K,K→° F,K→° C,° C→K

Unit(Imperial/US customary) input/result

thou,mil,po,P/,barley,in,li,ft,yd,rd,pole,perch,ch,fur,mi,lea,ftm,cb,NM

in$^2$,ft$^2$,yd$^2$,acre,mi$^2$,section,twp,rd$^2$,pole$^2$,perch$^2$,ch$^2$,rood

m.,fl s,fl dr,tsp,Tbsp,fl oz,gi,cp,pt,qt,gal,pk,bu,bbl,hogshead

in$^3$,ft$^3$,yd$^3$,acre-ft

gr,dr,oz,lb,st,qtr,cwt,ton,dwt,oz t,lb t

Unit(Other) input

Å,ua,l.y.,pc,xu

ha,are,b

L,t,carat,knot,Gal

dyn,kgf,lbf

atm,Torr,mHg,bar,mH$^2$O,psi

eV,calIT,calth,BtuIT

hp,PS,rpm

Poise,St,Gs,γ,Oe,Mx,sb,ph,Ci,R,rad(a),rem,g

hare,dare

Unit conversion

→meter,meter→,→meter$^2$,meter$^2$→,→meter$^3$,meter$^3$→,meter$^3$→in$^3$,→L,L→,L→in$^3$,→gram,gram→

Unit(Imperial/US customary) command

International,USSurvey,USArmy,UKVolume,USFluid,USBeer,USOil,USDry,USFDA,Canada,Austraria,

ShortMass,LongMass

MiYdFtIn,MiYdIn,MiFtIn,MiYd,MiFt,YdIn,FtIn,GalQtPtOz,GalPtOz,GalPt,GalOz,PtOz,

TonCwtLbOzDr,TonCwtLbOz,TonLbOz,TonLb,LbOz,StLb

Conversion function

→ : Add Units you want to convert left and right

Statistics input

X-data, Y-data, N-data : Substitute List

Statistics result memory

Σx, Σy, Σn, Σx2, Σxy, Σy2, Σx3, Σx2y, Σx4, x-ave, y-ave, δx, δy, sx, sy,
minx, maxx, medianx, Q1x, Q3x, modex, miny, maxy, mediany, Q1y, Q3y, modey, stat-a, stat-b, stat-c, stat-r

For Statistics data

statsort

Statistics function

x', y'

Statistics command

StatSINGLE, StatLINEAR, StatQUADRA, StatEULAR, StatGENERAL, StatPOWER, StatLOG, StatINVERSE

Financial calculation input/result

Fi-PV, Fi-FV, Fi-PMT, Fi-I%, Fi-N, Fi-C/Y, Fi-P/Y, Fi-Pay

Financial calculation function

→nom(, →eff(, bal(, Σprn(, Σint(, npv(, irr(

Probability density function

P(, Q(, R(
→Z

Complex number input/result

*i*, ∠

Complex number function

conj(, real(, image(, radius(, angle(
→rθ, →xy

Complex number result command

ComplexXY, ComplexRθ, NoComplex

List input/result

{, }

List function

culsum, difflist, sum, prod, sortA, sortD, mean, median, Q1, Q3, mode, stdDev, variant, abs_list
MIN(, MAX(, sum(, prod(, sortA(, sortD(, mean(, median(, Q1(, Q3(, mode(, stdDev(, variant(
dot_prod(, cross_prod(
Lback(, Lfront(

Matrix input/result

[[,][,]]

Matrix function

det, trans, identity, ref, rref

rnd-mat(, Rswap(, Radd(, Rmul(, Rmuladd(

Rback(, Rfront(, Cback(, Cfront(

Matrix command

Horizontal, Vertical : Switches matrix element direction

Matrix/List function

dim, fill(, aug(, getEle(, setEle(, Mat>List(, List>Mat(

Equation( : Solve equation

Formula input/result

X : Variable for Formula input

Formula function

Σ(, Π(, seq(, ∫(, dx(, Value(

Solver( : Solve formula

Polynomial expression/List function

Poly>List(, List>Poly(

Polynomial expression function

Vertex( : Get pole X-coordinate

For Result output

±, ?

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using RealNumber;

namespace ConsoleApp1 {

    class Program {

//  Length of a display buffer size
        private const int LINE_SIZE = 512;

        static void Main(string[] args) {

            UTF8byteArray OutputBuffer = new UTF8byteArray(LINE_SIZE);
            ErrorCode err_no;

            FunctionTable.FunctionAttributeTableCheck();

            while (true) {
                string InputString = Console.ReadLine();

                if (InputString == null)
                    break;

                if (InputString.Length >= LINE_SIZE - 1) {
                    Console.WriteLine("{0} -
> Error  Can't trest such large size.", InputString);
                    break;
                }

                if ((err_no = CalcExector.CalculatorExecute(ref InputString,
out ComplexStruct Num)) != ErrorCode.OK) {
                    Console.WriteLine("{0} -
> {1}", InputString, DispFormatter.errors[(int)err_no]);
                }
                else {
                    if (Num.type == PairType.REAL)
```

```
                        CalcExector.UnitTypeByMode(ref Num.real);
                    DispFormatter.PrintComplexValue(OutputBuffer, in Num);
                    OutputBuffer.count = OutputBuffer.index;
                    OutputBuffer.index = 0;
                    Console.WriteLine("{0} = {1}", InputString, OutputBuffer.
StringPart());
                }
            }
        }
    }
}
```