Big Integer Maths

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

```
' integer maths by TassyJim 07 Feb 2012

' MM Basic CAN do big integer maths!

' The included code uses integer maths to calculate
' 1, 1×2, 1×2×3, 1×2×3×4, etc. the successive factorial numbers.
' The limit in MM Basic is the length of a string or 255 digits.
' At this stage, only positive integers are allowed as inputs.
' This code is SLOW and I have not made any attempt to speed it up.
' Working in base 1000 (3 digits at a time) is the obvious way to
' gain speed but this exercise was a proof of concept rather than good code.
' I know a lot more about MM Basic subroutine now!
' It needs MM Basic V3.1 as it uses user defined subroutines.
' Converting to gosubs and line numbers is a task for someone else.

' Now I need to go the next step and allow negative numbers as inputs.

a$ = "13"
For i = 0 To 15 ' demo of the four operations
    add( a$ , Str$( i ),res$ )
    Print a$;" + ";i;" = ";res$
    min( a$ , Str$( i ),res$ )
    Print a$;" - ";i;" = ";res$
    multy( a$ , Str$( i ),res$ )
    Print a$;" * ";i;" = ";res$
    div( a$ , Str$( i ),res$ )
    Print a$;" / ";i;" = ";res$
    Print
Next i

a$="1"
For i = 1 To 55 ' printing up to factorial 55.
    b$=Str$(i)
    multy(a$, b$, res$)
    Print b$;"! = ";res$;"  (";len(res$);" )"
    a$=res$
Next i

End

Sub div( a$ , b$, c$ ) ' given a$ and b$, returns a/b in c$
    Local f, i, d, t, try$, nr$, a1$, b1$, a2$, z$
    a1$=a$ : b1$=b$ : c$="" ' make copies of a$ and b$ to preserve originals
    If Val( b$ ) = 0 Then   ' divide by zero
```

```
        c$ = "error"
    Else
        For i = Len( a1$ ) - Len( b1$ ) To 0 Step -1
            f = 0
            nr$ = "0"
            z$=String$(i,"0")
            For d = 0 To 9
                md( b1$ + z$ , d, try$ )
                big( a1$ , try$, t )
                If t Then
                    f = d
                    nr$ = try$
                EndIf
            Next d
            c$ = c$ + Str$( f )
            If f > 0 Then
                min( a1$ , nr$, a2$ )
                a1$ = a2$
            EndIf
        Next i
    EndIf
    trim (c$)
End Sub

Sub multy( a$ , b$, c$ ) ' given a$ and b$, returns the sum in c$
    Local i, h$, a1$, b1$, t$, d$, z$
    a1$=a$ : b1$=b$ : c$="" ' make copies of a$ and b$ to preserve originals
    If Len( b1$ ) > Len( a1$ ) Then ' swap number for greater speed
        h$ = a1$
        a1$ = b1$
        b1$ = h$
    EndIf
    For i = Len( a1$ ) To 1 Step -1
        z$=String$(Len( a1$ )-i,"0")
        md( b1$ , Val( Mid$( a1$ , i , 1 ) ) , t$)
        add( c$ , t$+ z$ ,d$)
        c$=d$
    Next i
    trim (c$)
End Sub

Sub md( a$ ,  nr, c$ ) ' multiply a$ by a single digit, result in c$
    Local hold, carry, i
    carry = 0 : c$=""
    For i = Len( a$ ) To 1 Step -1
        hold = Val( Mid$( a$ , i , 1 ) ) * nr + carry
        carry = Int( hold / 10 )
        c$ = Str$( hold Mod 10 ) + c$
    Next i
    If carry > 0 Then c$ = Str$( carry ) + c$
```

```
End Sub

Sub samelen (a$, b$) ' pads the shorter variable with leading zeros
    dif=Abs(Len( a$ )-Len( b$ ))
    If dif>0 Then
        z$=String$(dif,"0")
        If Len( a$ ) < Len( b$ ) Then
            a$ = z$+a$
        Else
            b$ = z$+b$
        EndIf
    EndIf
End Sub

Sub add( a$ , b$, c$ ) ' given a$ and b$, returns the sum in c$
    Local carry, hold, i,a1$, b1$
    c$="" : a1$=a$ : b1$=b$
    samelen a1$ , b1$
    For i = Len( a1$ ) To 1 Step -1
        hold = Val( Mid$( a1$ , i , 1 ) ) + Val( Mid$( b1$ , i , 1 ) ) +
carry
        carry = Int( hold / 10 )
        x$=Str$( hold Mod 10 )
        c$ = Right$( x$,1) + c$
    Next i
    If carry > 0 Then c$ = Str$( carry ) + c$
End Sub

Sub min( a$ , b$, c$ ) ' given a$ and b$, returns the difference in c$
    Local a1$, b1$, h$, i, hold, borrow
    c$="" : s$="" :a1$=a$ : b1$=b$ ' initialise variables and preserve a$
and b$
    samelen a1$ , b1$
    If a1$<b1$ Then ' if result is going to be negative, swap a1 & b1
        h$ = a1$
        a1$ = b1$
        b1$ = h$
        s$="-"
    EndIf
    For i = Len( a1$ ) To 1 Step -1
        hold = Val( Mid$( a1$ , i , 1 ) ) - Val( Mid$( b1$ , i , 1 ) ) + 10
- borrow
        borrow = 1- Int( hold / 10 )
        c$ = Str$( hold Mod 10 ) + c$
    Next i
    trim c$
    c$ = s$+c$ ' add the sign
End Sub

Sub big( a$, b$, biggest ) ' biggest = 1 if a$ >= b$
    Local a1$, b1$      'preserve a$ and b$
```

```
    a1$ = a$ : b1$ = b$
    biggest = 0
    samelen (a1$ , b1$)
    If a1$>=b1$ Then biggest = 1
End Sub

Sub trim (a$) ' given a$, leading zeros are stripped. a$ is altered.
    Local s$
    s$=""
    If Left$(a$,1)="-" Then ' if the number is negative, preserve the sign
        s$="-"
        a$=Mid$(a$,2)
    EndIf
    Do While Left$(a$,1)="0" ' strip leading characters while they are "0"
        a$=Mid$(a$,2)
    Loop
    If a$="" Then a$="0"
    a$=s$+a$ ' replace the sign
End Sub
```