Liberty Basic Day and Date Routines

LB Day-of-Week Conversions

Liberty BASIC is adept at handling various date string formats. It can handle 1- or 2-digit numbers for the month and day. It will also accept fully typed month names or 3-letter abbreviations. Years can be 1, 2, 3 or 4-digit numbers. Years from 0 to 99 are assumed as 2000 to 2099.
Some examples: Jul 4, 06 -- July 04, 2006 -- 7/4/6 -- 07/04/2006

In the LB universe, day 0 was Jan 1, 1901 and dates before 1/1/1901 return negative numbers.
Be aware that negative years are **incorrectly** handled as positive, so LB's Date$() function can not be relied upon for any date earlier than Jan 1, 100.

- -- -
  harmony

Liberty BASIC Day of Week

This routine uses LB's built-in Date$() function.

```
' Return Day-of-Week for dt$
function DayofWeek$(dt$)
  day = date$(dt$) mod 7 + 7  ' +7 for dates before Jan 1, 1901
  select case (day mod 7)
    case 0: d$ = "Tuesday"
    case 1: d$ = "Wednesday"
    case 2: d$ = "Thursday"
    case 3: d$ = "Friday"
    case 4: d$ = "Saturday"
    case 5: d$ = "Sunday"
    case 6: d$ = "Monday"
  end select
  DayofWeek$ = d$
end function
```

Zeller's Congruence for Day of Week

This version does not use LB's internal Date$() function, so the input parameters must be numbers. This routine might be prefered by LB 3.x users since it does not use the MOD operator.

```
' Day of Week using Zeller's Congruence
Function ZellerDoW$(yr, mo, dy)
  m = mo - 2
```

```
  if m<1 then
    m = m + 12
    yr = yr - 1
  end if
  c = int(yr/100)
  d = yr - 100*int(yr/100)
  f = int(2.6*m - 0.2) + dy + d + int(d/4) + int(c/4) + 5*c
  f = f - 7*int(f/7)
  dow$ = "Sunday Monday Tuesday Wednesday Thursday Friday Saturday"
  ZellerDoW$ = word$(dow$, f+1)
end function
```

Gregorian Year, Month, Day to Day of Week

```
' Gregorian date to Day-of-Week #
' This returns a number between 0 and 6 (0=Sun, 1=Mon, ...)
function G2dow(yr,mo,dy)
  a = int((14-mo)/12)
  m = mo + 12*a - 2
  y = yr - a
  G2dow = (dy + y + int(y/4)-int(y/100)+int(y/400)+int(31*m/12)) mod 7
end function
```

Gregorian Year, Month, Day to Julian Day

```
' Convert Gregorian year,month,day to Julian Day #
function G2Jd(yr,mo,dy)
  a = int((14-mo)/12)
  y = yr + 4800 - a
  m = mo + 12*a - 3
  G2Jd = dy + int((153*m+2)/5)+365*y+int(y/4)-int(y/100)+int(y/400)-
32045
end function
```

Julian Day to Gregorian Year, Month, Day

```
' Julian day to Gregorian date
function Jd2G(jd, byref yr, byref mo, byref dy)
  a = jd + 32044
  b = int((4*a+3)/146097)
  c = a - int((146097*b)/4)
  d = int((4*c+3)/1461)
  e = c - int(1461*d/4)
  m = int((5*e+2)/153)
  dy = e - int((153*m+2)/5) + 1
```

```
  mo = m + 3 - 12*int(m/10)
  yr = 100*b + d - 4800 + int(m/10)
end function
```

Note for LB 3.x users

LB 3.x does not have the MOD operator, so you will need to provide one for some of these functions.

```
' LB mod function for LB 3.x users
function LBmod(a, b)
  LBmod = a - b * int(a/b)
end function
```

Example Usage
For the DayofWeek$() function, replace the two lines using MOD with these:

```
  day = LBmod( date$(dt$), 7) + 7  ' +7 for dates before Jan 1, 1901
  select case LBmod(day, 7)
```

Additional Information

The calculations for the Gregorian date and day routines are from Claus Tøndering's excellent Calendar FAQ page at http://www.tondering.dk/claus/calendar.html
They can be freely distributed for all non-commercial uses.