

```

// ***** START DATE RANGE TEST CODE ***** //
//
//   When running this code, place breakpoints where indicated with the text as shown
//   in order to provide formatted text to the Debuggers 'Output' window while Debugging...
//
// Run from 2026 to 2096 (126 to 196), Changes 'EpochAlypse' date of Jan 19, 2038 to Jan 19, 2094 (another 56 years).
// How it works: If the year is later than 2035 (i.e., 2036 and later) subtract 56 from the year. This causes 'mktime'
// to go back in time to 1980, as the calendar from 1980 onward mimics the calender from 2036 and onward in all respects
// except for the year of course. So, after performing the 'mktime' calculations, add 56 back to the year in order to
// correctly display to the user the current year on the LCD GUI Interface. So what's happening is that we're 'faking out'
// the mktime function in order to get another 56 years of life past the currrent 'EpochAlypse' of the year 2038...
static volatile struct tm timetst;
time_t seconds, secondsprior;
int Yr, xYr, Mn, Day, DayMax;
    Yr = 126;    // Start test in 2026...
    Mn = -1;     // Just to start, then always 0-11
int Epochalypse = 0;
int LeapYr = 0;

    // Run test up to 2096, illustrating how the dates are completely wrong after the 'new'
    // EpochAlypse date of Jan 19, 2094 and the year becomes 1957 and onward past this date.
while(Yr < 196){
    do{
        if(Mn < 11)
            Mn += 1;
        else
            Mn = 0;
        Day = 0;

        do{
            Epochalypse = 0;
            if(Yr > 135) Epochalypse = 1;

            Day++;
            switch(Mn){
                // These months have 31 days
                case 0:    // January
                case 2:    // March
                case 4:    // May
                case 6:    // July
                case 7:    // August
                case 9:    // October
                case 11:   // December
                    DayMax = 31;
                    if(Day > 31)
                        Day = 0;

                    break;

                // These months have 30 days
                case 3:    // April
                case 5:    // June
                case 8:    // September
                case 10:   // November
                    DayMax = 30;
                    if(Day > 30)
                        Day = 0;

                    break;

                case 1:    // February - and this month, depending on leap year, have either 28 or 29 days
                if ( (((Yr-100) % 4) == 0) && (((Yr-100) % 100) != 0) || (((Yr-100) % 400) == 0) ) //leap year
                {
                    DayMax = 29;
                    if(Day > 29)
                        Day = 0;
                }
            }
        }
    }
}

```

```

        }
        else
        {
            DayMax = 28;
            if(Day > 28)
                Day = 0;
        }
        break;

        default:
        break;
    }

    // Just to have a place to output "***** LEAP YEAR *****" on breakpoint...
    if((Mn==1) && (Day==29)){
        LeapYr=1;
    }
    LeapYr=0;

    timetst.tm_hour = 0;
    timetst.tm_min = 0;
    timetst.tm_sec = 0;
    timetst.tm_mon = Mn;
    timetst.tm_mday = Day;
    timetst.tm_year = Yr;
    if(Epochalypse==1){
        timetst.tm_year-=56; // if we're in the 'EpochAlypse' (years past 2035), head back to the 80's and forward...
    }
    timetst.tm_isdst = -1;

    seconds = mktime(&timetst);
    secondsprior = seconds - 60;
    timetst = *localtime(&seconds);

    // Adjust year back to 'reality' for the Breakpoint output 'Test Date' text to be 'correct'...
    if(Epochalypse == 1){
        timetst.tm_year+=56;
    }

Breakpoint:    timetst = *localtime(&secondsprior);    //Place Breakpoint here: "Test Date: Month:{timetst.tm_mon+1}, Day:{timetst.tm_mday}, Year:{timetst.tm_year+1900} - Midnight"
    // Adjust year back to 'reality' for the Breakpoint output 'Date Prior' text to also be 'correct'...
    if(Epochalypse == 1){
        timetst.tm_year+=56;
    }

Breakpoint:    }while(Day < DayMax);    //Place Breakpoint here: "Date Prior: Month:{timetst.tm_mon+1}, Day:{timetst.tm_mday}, Year:{timetst.tm_year+1900} - Minute Prior"
    if(Mn == 11)

Breakpoint:        Yr += 1;    //Place Breakpoint here: "YEAR-YEAR-YEAR-YEAR-YEAR-YEAR-YEAR-YEAR-YEAR-YEAR-YEAR"

Breakpoint:    }while(Mn < 11);    //Place Breakpoint here: "-MONTH-MONTH-MONTH-MONTH-MONTH-MONTH-MONTH-MONTH-MONTH-MONTH-"
    };
    // ***** END DATE RANGE TEST CODE ***** /
```