Julian Day C Code

Here is a snippet of C code to convert a day, month and year in the Gregorian calendar in to a Julian Day number.

```
long gregorian_calendar_to_jd(int y, int m, int d)
{
    y+=8000;
    if(m<3) { y--; m+=12; }
    return (y*365) +(y/4) -(y/100) +(y/400) -1200820
            +(m*153+3)/5-92
            +d-1
    ;
}
```

A similar function works for the Julian calendar; just omit the last two year-related corrections and compensate for the epoch with a different constant.

Notes

`y+=8000;`

> The most negative year in the current Julian period is -4713 Gregorian (4714 BC), so we want to add something to `y` so we can simplify reasoning about our calculations by considering only positive year values. Any value divisible by 4000 could be chosen, since the pattern of leap years in the reformed Gregorian calendar repeats every 4000 years.

`if(m<3) { y--; m+=12; }`

> Since leap days are added at the end of February, it makes sense to consider January and February as the last two months (numbered 13 and 14) of the previous year. That way, leap days are always added at the end of the notional year.

`return (y*365) +(y/4)`

> The Julian calendar has normal years of 365 days, plus extra leap days every fourth year.

`...-(y/100) +(y/400)`

> The Gregorian calendar adds the correction that a century year is not a leap year, unless it is also divisible by 400.

`...-1200820`

> This correction allows for the 3287-year difference between th= arbitrary positive year offset we added earlier and the real Julian Day epoch of 1st. January, 4713 BC, Julian calendar (24th. November, 4714 BC, Gregorian calendar).

`...+(m*153+3)/5-92`

> With the months starting with March=3 and ending with February=14, there is a simple pattern of month lengths: 31, 30, 31, 30, 31, 31, 30, 31, 30, 31, 31 That is, there are always 30=150/5 days in each month with an additional day every 3/5 month. Since we started with March=3, the first value given would be (3*153+3)/5 = 92, but we want to start from zero so we make a correction.

`...+d-1`

> The Julian Day number goes up once a day. Days of the month are conventionally numbered from 1, so we subtract 1.

This code is used to drive the calendar conversion form.