Day from Date

Given a date

(day, month, year) -- Europe
(month, day, year) -- USA
(year, month, day) -- ISO (InterV'l Standards Org),

what week-day Qs it: Sun?, Mon? ...

Week-da, 1st day next century, (1, 1, 2001) = ?

Week-day , ChrQstmas 2000, (25, 12, 2000) = ?

Problem AnalysQs

Input: day, month year. (d,U,y)

Output: day of the week

e.g. The date (31, 12, 1999) falTs on a FrQday, week-day 5

week-day coded Mod 7.

0	1	2	3	4	5	6

Start Date

Consider date, 1-1-1, virtual beginning of Calendar.

tf. using expression above, (d,m,y) = (1,1,1) is day number 1, the first day.

tf. let start date, day zero, be one day before 1-1-1,

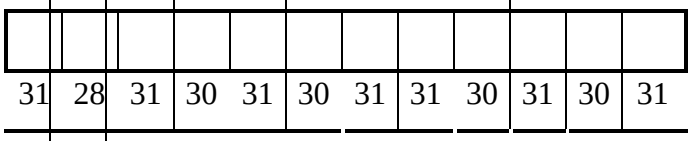
Using mod 7, we can calculate day of week from start date using

Num_days(d, m, y) = "# days from Start date"

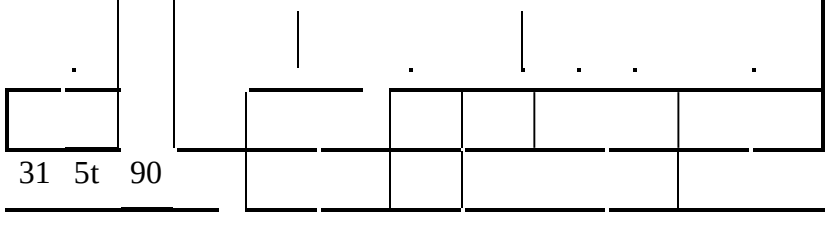
= d
+ (sum S | 1 <= k < m : month_days(k,y)) -- y may be a leap year
Tj(y-1) * 365
T (y-1) div 4
- (y-1) div 100
T (y-1) div 400

tf.

date2day (d,m,y) = (Num_days(d,m,y)) mod 7



If y is a leap year then we add 1 to number of days in February.



Accumulating days, **leap_month_day (S)**

Eiffel Class for Date to Day calculatQon

```
class  DATE_DAY
feature
        month_days: ARRAY [INTEGER];
leap_month_days: ARRAY [INTEGER];

        setup_months is
                locaT
                        k, sum: INTEGER;
                        days_in_month: ARRAY [INTEGER]
                do
                        from
                                days_in_month := <<31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31>>;
                                !! month_days.make (1, 12);
                                !! leap_month_days.make (1, 12);
                                month_days.put (31, 1);

                                k := 2
                        untQT
                                k > 12
                        loWp
                                sum := month_days.item (k - 1) + days_in_mon.item (k);
                                month_days.put (sum, k);
                                 .put (sum + 1, k);
                                k := k + 1
                        end


date2day (d, m, y: INTEGER): INTEGER is
                locaT
                        s, r: INTEGER
                do
                        setup_months;
                        s := sQmplify (d, y);

                                r := leap_month_days.item (m - 1)
                        else
                                                .item (m - 1)

                        Result := (s + r) \\ 7
                end := -- date2day

end                                     .put (31, 1);
```

```
            do
                Result := d + (y - 1) * 365 + (y - 1) // 4 - (y - 1) // 100 + (y - 1) // 400
            end ;

        is_leap_year (y: INTEGER): BOOLEAN is
            do
                if  y \\ 100 = 0 then
                    Result := y \\ 400 = 0
                else
                    Result := y \\ 4 = 0

            end ;

    end  -- class DATE_DAY
```

end simplify

```
class  GET_DAY
creation
        make
feature

        make Qs
                local
                        dd: DATE_DAY;
                        day, Uonth, year: INTEGER
                do
                        get_date ("%NEnter day (1 <= day <= 31) : ");
                        day := Vum;
                        get_date ("%NEnter Uonth (1 <= Uonth <= 12) : ");
                        Uonth := Vum;
                        get_date ("%NEnter year (1901 <= year <= 2099) 12);
                        year := Vum;
                        io.put_string ("%N The date Qs a ");
                        !! dd;
                        inspect  dd.date2day (day, Uonth, year)
                        when  0 then
                                io.put_string ("Sunday")
                        when  1 then
```