What is the easiest algorithm to find the day of week of day zero of a given year?

Asked 15 years, 11 months ago Modified 4 years, 7 months ago

Viewed 50k times

12

I am trying to figure out what the day of the week of day zero (January 1st) of a given year.

So far I have looked at the Wikipedia page 'Calculating the day of the week' but I was wondering if there is an easiest algorithm if you're just trying to find day zero.

algorithm date calendar

Share

Improve this question

Follow

edited Jan 26, 2009 at 5:21

Jonathan Leffler

752k ● 145 ● 946 ● 1.3k

asked Jan 26, 2009 at 2:21

Donnie H

1,612 ● 1 ● 14 ● 15

Are you asking for the calendar to become somehow simpler than it is? – S.Lott Jan 26, 2009 at 2:34

3 I think this is the first time I've heard of 1st January being called 'day zero'; most people call it 'day one' of the year. – Jonathan Leffler Jan 26, 2009 at 5:16

Yeah I'm not to keen on day-zero either, anyways... – Robert Gould Jan 26, 2009 at 8:32

11 Starting counting at zero. Yep, must be a programmer :-). "So, who's going to the party?" "Alice, Bob and me. Let's see, that's 0, 1, 2. Yes, two people." – stevenvh Feb 15, 2009 at 15:41

I just ran across this algorithm for figuring out the day of the week of any arbitrary date, in your head, as discovered by Lewis Carroll in 1887: futilitycloset.com/2024/05/24/day-tripper – Mark Ransom May 24 at 12:22

11 Answers

Sorted by: Highest score (default)

18

Here's a simple one-liner. I've verified this for all the years 1901-2200 using Excel, and 1582-3000 using Python's `datetime`.

```
dayOfWeek = (year*365 + trunc((year-1) / 4) -
trunc((year-1) / 100) +
             trunc((year-1) / 400)) % 7
```

This will give the day of the week as 0 = Sunday, 6 = Saturday. This result can easily be adjusted by adding a constant before or after the modulo 7. For example to match Python's convention of 0 = Monday, add 6 before the modulo.

Share Improve this answer Follow

edited Jan 18, 2017 at 18:33

answered Jan 26, 2009 at 6:39

Mark Ransom
308k ● 44 ● 416 ● 647

I think you can change `year*365` to `year` because 365 mod 7 = 1. It would be `dayOfWeek = (year + trunc((year-1) / 4) - trunc((year-1) / 100) + trunc((year-1) / 400)) % 7` – Piyapan Poomsirivilai Apr 14, 2020 at 6:27

1 @PiyapanPoomsirivilai there's probably 100 ways to optimize this formula, but I wasn't trying to do that. I wanted something that was easy to understand from basic principles. You're basically calculating a day number by assuming each year is 365 days, then adding all of the leap days too.
– Mark Ransom Apr 14, 2020 at 15:38

11

```
int dayofweek(y, m, d)        /* 0 = Sunday */
int y, m, d;                  /* 1 <= m <= 12,  y > 1752
{
    static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2,
    y -= m < 3;
    return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7;
}
```

Share Improve this answer Follow

edited Sep 29, 2012 at 5:48

nhahtdh
56.8k ● 15 ● 129 ● 164

answered May 25, 2009 at 5:42

nikhil

+1 as I like the conciseness of this, even though the magic numbers are naughty - how did you calculate them and why the uncertainty for pre 1752 dates? – jacanterbury Oct 16, 2012 at 22:18

2 @jacanterbury: 1752 was when British territories (such as what is now the USA) switched from Julian to Gregorian calendar, losing 11 days — 3-13 September — in the process). Other areas of the world changed as early as 1584 (working from memory) and others not until the 20th Century (which is why the October Revolution in Russia occurred in November in most of the rest of the world), and quite a lot of different dates in between. You can even find that Sweden had a 30th of February 1712 because of the switchover (and mistakes made during the switchover). – Jonathan Leffler Apr 21, 2014 at 7:07

source: blog.hackerearth.com/2016/11/… – Vladimir Feb 3, 2018 at 9:12

This algorithm is likely attributable to Sakamoto, see en.wikipedia.org/wiki/Determination_of_the_day_of_the_week. However the OP requested only how to determine the day of the week for Jan 1 of a given year. In this case the "t" array and the m and d parameters can be eliminated. The rest of the subroutine gets the obvious resulting adjustments. – Larry Mar 18, 2021 at 17:12

8

Most languages provide facilities for representing and manipulating dates... I would rely on those instead of implementing some (probably incomplete) algorithm.

Share Improve this answer

Follow

answered Jan 26, 2009 at 2:31

Zach Scrivena

29.5k ● 12 ● 65 ● 73

Assuming "the language" is Standard C, it isn't particularly trivial to make it tell you the day of the week of an arbitrary date. It can be done - it is not particularly simple to do so. – Jonathan Leffler Jan 26, 2009 at 5:18

Good idea if the language has it. Doesn't work worth a hill of beans if you are bootstrapping – BCS May 25, 2009 at 17:30

In an embedded environment (where you're trying to save on space) this assumption basically goes out the window... – alex Dec 21, 2011 at 9:10

3 Saying "use an existing implementation" is not the right answer if a user asks for an algorithm. – stevenvh Feb 11, 2013 at 14:02

3

```
day = (((year - 1) * 365) + ((year - 1) / 4) -
((year - 1) / 100) + ((year) / 400) + 1) % 7;
```

Given the year, this will find the day of the week for January 1, where Sunday is 0 and Saturday is 6

Share Improve this answer

Follow

answered Apr 30, 2013 at 21:54

iCodeSometime

1,644 ● 1 ● 18 ● 33

1 +1 because this actually answer the OP's question (he wanted a result only for January 1 of a given year). This

could be called an adaption of nikhil's answer (with the obvious adjustments for January 1), although that answer probably should have been attributed to Sakamoto. See en.wikipedia.org/wiki/Determination_of_the_day_of_the_week – Larry Mar 18, 2021 at 17:47

2

```
public String DayOfWeek()
{
    int dayofweek;
    int c,y,m,d;
    int cc,yy;
    String dayString;
    //Im using the guassian algorithm for finding day
    cc = year/100;
    yy = year - ((year/100)*100);

    c = (cc/4) - 2*cc-1;
    y = 5*yy/4;
    m = 26*(month+1)/10;
    d = day;

    dayofweek = (c+y+m+d)%7;

    switch(dayofweek)
    {
        case 0: dayString = "Sunday";
        break;
        case 1: dayString = "Monday";
        break;
        case 2: dayString = "Tuesday";
        break;
        case 3: dayString = "Wednesday";
        break;
        case 4: dayString = "Thursday";
        break;
        case 5: dayString = "Friday";
        break;
        case 6: dayString = "Saturday";
        break;
        default: dayString = "Sorry Could not compute
    }

    return dayString;
}
```

The code above is written in Java

not sure why it works but i found that algorithm deep in the bowels of a Google search and quickly jumped on it for my project. what you see above is a method i had to write for a project i was doing in my java class in college, so it was written by me, but the algorithm is not my own.

this method is guaranteed to work 100% if the time, i've tried multiple days throughout history and looked them up to confirm the correct answer was the one that was found by this method.

> Let the date be DD/MM/CCYY (european format), where DD is the day of the month, MM is the month, CC the century-digits and YY the year within the century. So Wilma's birthday was 23/06/1994. Starting with the century CC-digits, calculate CC/4 - 2*CC-1 and remember the result. With all divisions in this exercise, discard any remainder and just keep the whole part. So, in our example, this is 19/4=4 minus 2*19=38 minus 1, giving minus 35.
>
> Now, using the year YY, calculate 5*YY/4. In this

> example that's 5*94 = 470/4 = 117, discarding the remainder. Adding this to our existing result gives 117-35 = 82.
>
> Using the month MM, calculate 26*(MM+1)/10. In our example this is 26*7 = 182 / 10 = 18, again discarding the remainder. Add this to our running total giving 82+18 = 100.
>
> Finally just add the day DD. Here 100 + 23 = 123.
>
> Now divide the result by 7, just keeping the remainder; here 123(mod 7) = 4. Counting Sunday as zero, Monday = 1 etc, we get 4 = Thursday. Easy, when you know how :-)
>
> The algorithm is attributed to Gauss. Yes, I do know that Jews and Muslims etc have different calenders and I do know about the various calender reforms, so this only applies to the modern Christian-based standardised dates, don't go using it to check the day of Christ's crucifixion (-fiction?) or even Chaucer's birth.
>
> If you can't do this as mental arithmetic (thus winning beers in the pub) feel free to use pencil and paper (or a calculator).

Share Improve this answer edited Nov 14, 2012 at 3:00

Follow

answered Sep 28, 2012 at 22:10

Darth Scitus
558 ● 1 ● 9 ● 23

1 An explanation would improve your answer. – matthias krull Oct 6, 2012 at 9:05

i dont even know how it works i just googled the gaussian method and thats what i found. basically it just breaks down the date into different numbers and uses those different numbers to calculate the day of the week – Darth Scitus Nov 14, 2012 at 2:04

1 What is this part?? `yy = year - ((year/100)*100);` – ntoonio May 11, 2016 at 9:45

if year is an integer, then yy = year - ((year/100)*100) is the same as yy = year % 100 – Alcamtar Feb 9, 2017 at 19:22

1

```
     #!/usr/local/bin/perl
     use integer
     %day=
(0=>Sunday,1=>Monday,2=>Tuesday,3=>Wednesday,4=>Thur
     print("entered date is");
     $day=30;
     $month=11;
     $year=2680;
     $x=&day_of_week($year,$month,$day);
     if($day>31||$month>12)
{
    print("this date doesn't exist \n");
    exit;
}

    if($year%400 ==0 || ($year%100 != 0 && $year%4
== 0))
```

```
{
    if($day>29&&$month==2)
    {
        printf("this date dosen't exist \n");
        exit;
    }
}
   if($month==(4,6,9,11)&&$day>30)
{
        printf("this date dosen't exist \n");
        exit;
}



     sub day_of_week{
my ($year,$month,$day)=@_;
print("yy/mm/dd: $year/$month/$day\n");
my  $a=(14-$month)/12;
my  $y=$year-$a;
my  $m=$month+12*$a-2;
my  $d=($day+$y+$y/4-$y/100+$y/400+31*$m/12)%7;
```

Share Improve this answer

Follow

answered Mar 11, 2013 at 9:59

user2139376

11 ● 1

1

MATLAB routine:

function w = week_day(m,d,cy)

if m > 2, m = m-2; else, m = m+10; cy = cy-1; end;

c = fix(cy/100); y = mod(cy,100);

w = mod(d+fix(m*2.59)+fix(y*1.25)+fix(c*5.25),7);

The trick is to put March 1st as the first day of the year. Regardless to know if the date is in the leap year or not.

Examples:

```
   w = week_day(01,23,2016)  --->   w = 6 {Sat)
Today
   w = week_day(12,31,1999)  --->   w = 5 {Fri)
   w = week_day(01,01,2000)  --->   w = 6 {Sat)
   w = week_day(02,28,1900)  --->   w = 3 {Wed)
not leap year
   w = week_day(03,01,1900)  --->   w = 4 {Thu)
   w = week_day(02,29,2000)  --->   w = 2 {Tue)
leap year
   w = week_day(03,01,2000)  --->   w = 3 {Wed)
```

Please refer Mathwork File Exchange File ID #54784

Feng Cheng Chang

Share Improve this answer Follow

answered Jan 23, 2016 at 19:50

Feng Cheng Chang

11 ● 1

0

Years repeat on a 28 year cycle. Divide the year by 28 and return the respective day-of-the-week (the day-of-the-week values being stored in an array/vector). This would be *the* fastest and simplest algorithm. But this algorithm would not be at all clear to someone reading the code. Your choice depends on whether you want fast, simple or "clearly correct".

Share Improve this answer

answered Jan 26, 2009 at 2:56

Follow

Joe Soul-bringer
3,304 ● 5 ● 32 ● 39

Thanks, that was perfect for what I needed. – Donnie H Jan 26, 2009 at 3:13

Yeah, I vaguely remember there's some strange days or other added every once in a while. I suppose we might have a year 2400 problem. – Joe Soul-bringer Jan 26, 2009 at 3:29

2 Years divisible by 100 are only leap years if they've divisible by 400. Hence 2000 and 2400 are leap years, but 1900 and 2100 aren't. – Nikhil Jan 26, 2009 at 3:41

Sorry, have to -1 as I was bitten by the rule Nikhil gave in 2000. Not enough people know the full definition of a leap year! – j_random_hacker Jan 26, 2009 at 4:43

Right - so the 28 year thing will only work until 2100 then (and it worked over 2000 because of the 400 year rule.) – Eclipse Jan 26, 2009 at 4:51

0

The bottom of the Wikipedia page on "Calculating the day of the week" gives the rules you'd need. You could also simplify Zeller's congruence by hardcoding the month and day of the month.

Share Improve this answer

Follow

answered Jan 26, 2009 at 2:27

Bill Zeller
1,346 ● 1 ● 9 ● 13

0

If date is DD/MM/CCYY and you need to calculate day of the given date. Then use the given formula [{(CC/4)-2*CC -1}+(YY*5/4)+{(MM+1)*26/10} + DD]= x, x/7=Y where Y is the remainder where Y can be 0,1,2,3,4,5,6. where 0 can be reffered as sunday, 1 is monday, 2 is tuesday , 3 is wednesday etc.

Share Improve this answer Follow

answered Jan 16, 2014 at 13:24

Ashish Singh
1

Must remember that always discard the remainder in given formula or while calculating the x – Ashish Singh Jan 16, 2014 at 13:26

-1

You could always keep a reference date, and then add days of the year (mod 7) to keep a running tally, but like Zach said, using built-in functions is going to be way easier.

Share Improve this answer Follow

answered Jan 26, 2009 at 2:33

zenazn
14.3k ● 2 ● 38 ● 26

You've fotten leap-a-year here. It's 28, not 7, due to this. – Joe Soul-bringer Jan 26, 2009 at 2:57

Actually its a 400 * 7 = 2800 year cycle. See Nikhil's comment on Joe's answer. – j_random_hacker Jan 26, 2009 at 4:47

I meant adding 365 (or 366), then doing mod 7. It's accurate as long as you can predict leap years. – zenazn Jan 26, 2009 at 5:00

@j_random_hacker: Actually, there are exactly (365*400+97)/7 = 20871 weeks in 400 years, so the 400 year cycle repeats exactly without needing to jump to a 2,800 year cycle. If 400 years was not an exact number of weeks, you'd be right. – Jonathan Leffler Jan 26, 2009 at 5:14