Search…

Tricks to Solve Calendar Problems

Last Updated : 02 Apr, 2025

The topic of **calendars** is essential for various competitive exams and placement tests. While it might initially seem challenging, the tricks and methods shared here will simplify the concepts, making calendars easy to understand and apply.

A **calendar** is a system for organizing and dividing time into days, weeks, months, and years. **Calendars** are based on astronomical events, such as the rotation of the Earth (**day)**, the lunar cycle (**month**), and the Earth's orbit around the sun (**year)**, and have been essential for managing time and planning in societies throughout history.

Basic Concepts and Terminology

Leap Year:

- A year with 366 days, in which February has 29 days.
- A year is a leap year if it is divisible by 4. However, if it's a century year (ending in 00), it must be divisible by 400 to be a leap year.
- For example, 2000 was a leap year, but 1900 was not.

Ordinary Year

- A year with 365 days, in which February has 28 days.
- Non-leap years are not divisible by 4 or do not satisfy the century rule (not divisible by 400).

- The number of days left after calculating complete weeks in a given period.
- Odd days help in determining the day of the week for a given date.
- For example, if 10 days pass, it's 1 complete week (7 days) and 3 odd days.

Day Codes

Days of the week can be assigned numerical codes to simplify calculations:

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Satu
0	1	2	3	4	5	

Month Codes

Each month is given a code to simplify day calculations, particularly for non-leap and leap years:

Month	Code
January	For leap year: 6, For non-leap year: 0
February	For leap: 2, For non-leap year: 3
March	3
April	6

Month	Code
June	4
July	6
August	2
September	5
October	0
November	3
December	5

Century Codes

The century codes are used to simplify day-of-week calculations for dates across centuries.

Century	Code
1600-1699	6
1700-1799	4

Century	Code
1900-1999	0
2000>=	6

Types of Questions

Type 1: Given a month, date, and year, use this data to find the day of the week.

Type 2: Given a day, with its date, find which day will be after **x months**, **x days**, or **x years.**

Tricks and Shortcuts for Calendar Questions

Let us Understand them with the help of small and easy tricks.

The Date-to-Day Code Trick

Suppose we have given a date like 25 May 2003. and we have to calculate the day for this

Step: 2 = month code

Step: 3 = Date

Step: 4 = last 2 digits of year divided by 4(quotient)

Step: 5 = Year code

Now add all these and find its remainder with 7.

Now, whatever the value you are getting convert it into Week Day Code and We will get the required Day.

Solving the above example:

- *Step: 1* The last 2 digits of a year is 03 i.e **3.**
- *Step: 2* month code for May is **1.**
- *Step: 3* Date is **25.**
- *Step: 4* The last 2 digits divided by 4 is 0 i.e 3/4 gives quotient as **0.**
- *Step: 5* Year code is **6.**

Adding them (values obtained after all the 5 steps)

We get (3 +1 + 25 + 0 + 6) = 35.

So, **35/7** = 0(Remainder).

And we have already mentioned in Trick's section that 0 is the Code for Sunday. So, the Weekday on 25 May 2003 was Sunday.

Day Difference Trick

It was Monday 15th October 1923. What was the day on 17 November 1923?

TRICK - We simply have to add the day's between two given dates, and add it to the code of the day given
in question and divide it by 7 to get the remainder .

Since October has 31 days, there are 16 days from October 15 to October 31, and 17 days from November 1 to November 17.

Adding these gives (16 + 17) = 33 days.

Dividing 33 by 7, we get a remainder of 5.

Now, add 5 to the code for Monday (1, as given in the tricks), then divide by 7 to find the remainder: (5 + 1) mod 7 = 6. This remainder of 6 corresponds to Saturday.

*Hence, it was **Saturday** on 17ᵗʰ November 1923.*

Zeller Congruence

Zeller's congruence is an algorithm devised by Christian Zeller to **calculate the day of the week.**

It is given by the formula:

$$f = \left(d + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + Y + \left\lfloor \frac{Y}{4} \right\rfloor + \left\lfloor \frac{C}{4} \right\rfloor - 2C\right) \mod 7$$

Where

- **f** is the day of the week.
- **d** is the day of the month.
- m is the month (March = 3, April = 4, ..., December = 12; January = 13, February = 14 of the previous year).
- **Y** is the **year mod 100.**
- **C** is the **year / 100.**

In the traditional Zeller's congruence algorithm, the resulting value, f, is mapped as follows:

- 0 = Saturday
- 1 = Sunday
- 2 = Monday

- 5 = Thursday
- 6 = Friday

For example: 20th March 1882

Solution:

For the given question the solution through Zeller's formula is :

- *f is to be calculated*
- *d = 20*
- *m = 3*
- *Y = 82*
- *C = 18.82 = **18** (we will take **18** as we have to choose the floor value).*

$$f = \left(20 + \left\lfloor \frac{13(3+1)}{5} \right\rfloor + 82 + \left\lfloor \frac{82}{4} \right\rfloor + \left\lfloor \frac{18}{4} \right\rfloor - 2 \times 18\right) \mod 7$$

= (20 + 10 +82 + 20 +4 -36) mod 7

(100) mod 7

= 2 (According to the code for the week).

Hence it was Monday on 20th march 1882.

The Ordinary Year Month Code Trick

In an ordinary year, the first days of the month are the same for:

- *January and October (1-10)*
- *April and July (4-7)*
- *February, March, and November (2-3-11)*
- *September and December (9-12)*

The Leap Year Month Code Trick

- *January, April, and July (1-4-7)*
- *February and August (2-8)*
- *March and November (3-11)*
- *September and December (9-12)*

Also Check:

- **Calendars Aptitude Questions & Answers**
- **Calendars Practice Quiz**

| Comment | More info | Advertise with us |

Similar Reads

1. Teaching Calendar Skills to Kindergarten
2. Calendars
3. Calendar - Aptitude Questions and Answers
4. Puzzle 51| Cheryl's Birthday Puzzle and Solution
5. Find the Probability of getting 53 Sundays in a Non-Leap Year?
6. Find whether a day fall under the GEEK year
7. Puzzle 23 | (Days of month using 2 dice)
8. Puzzles MCQ
9. Python program to print Calendar without calendar or datetime module
10. Print calendar for a given year in C++

(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305

Advertise with us

Company

About Us

Legal

Privacy Policy

Careers

In Media

Contact Us

Corporate Solution

Campus Training Program

Explore

Job-A-Thon

Offline Classroom Program

DSA in JAVA/C++

Master System Design

Master CP

Videos

Tutorials

Python

Java

C++

PHP

GoLang

SQL

R Language

Android

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

DSA Interview Questions

Competitive Programming

Data Science & ML

Data Science With Python

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Python Tutorial

Python Examples

Django Tutorial

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Computer Science

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Databases

SQL

MYSQL

PostgreSQL

PL/SQL

MongoDB

Preparation Corner

Company-Wise Recruitment Process

Aptitude Preparation

Puzzles

Company-Wise Preparation

More Tutorials

Software Development

Software Testing

Product Management

Project Management

Linux

Excel

All Cheat Sheets

Courses

IBM Certification Courses

DSA and Placements

Web Development

Data Science

Programming Languages

Programming Languages

C Programming with Data Structures

C++ Programming Course

Java Programming Course

Python Full Course

AWS Solutions Architect Certification

Salesforce Certified Administrator Course

GATE DA Rank Booster

GATE CS & IT Course - 2026

GATE DA Course 2026

GATE Rank Predictor

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy