Tarun Eswar, Vaishnavi Harish, Isabella Palit, Rianna Santra

POW 1: Happy Birthday

Problem Statement:

Many people look to the day of the week they were born in as an indication of their personality. Therefore, we intended to design an algorithm capable of determining the day of the week on which someone was born. Leap years and an inconsistent amount of days in each month makes the problem more complicated, but we were able to determine multiple algorithms capable of solving the aforementioned problem.

Givens:

We were given the October 2022 Calendar, as well as many test dates. We were also given the number of days per month.

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Day	31	28/9	31	30	31	30	31	31	30	31	30	31

A quick note:

All of the methods utilize the operator mod, which is a mathematical technique that simplifies the calculations below. To demonstrate how mod manipulates numbers:

If a / b = c with a remainder of d, then a mod b ≡ d

Ex: 95/7 = 13 with a remainder of 3, so 95 mod 7 ≡ 3
Ex: 4/9 = 0 with a remainder of 4, so 4 mod 9 ≡ 4

Process:

Process 1 (Tarun):

Based on the October Calendar, the first day of each month in 2022 was found by manual calculations. These were done by shifting the values of 1 - 31 or 1 - 30 left or right amount of spaces using the given October calendar given. After computing the values of the first day of each month for 2022 and 2023, a common pattern emerged:

Let:

V_{MN} = Value added to output the index of first day for next month for a non leap year
V_{ML} = Value added to output the index of first day for next month for a non leap year
V_Y = Value added to output the index of first day for next year

Table I. (*Table is in mod 7)*

	V_{MN}	+3	+1	+3	+2	+3	+2	+3	+3	+2	+3	+2	
	V_{ML}	+3	+0	+3	+2	+3	+2	+3	+3	+2	+3	+2	
V_Y	Y/M	Jan	Feb	Mar	April	May	June	July	Aug	Sept	Oct	Nov	Dec
-1	2023	0	3	3	6	1	4	6	2	5	0	3	5
-1	2022	6	2	2	5	0	3	5	1	4	**6**	2	4
-2	2021	5	1	1	4	6	2	4	0	3	5	1	3
-1	2020	3											
-1	2019	2											
-1	2018	1											
-2	2017	0				Cells follow same pattern and are not needed							
-1	2016	5											
-1	2015	4											
-1	2014	3											
-2	2013	2											
	2012	0											

Strategy:

Step 1: Find the first day of the year

Let:

d = given day
m = given month
y = given year
z = first day of a given year
l = number of leap years

Based on the table above, the first day of the next year will be 1 index value less than the previous if the next year is not a leap year. If the next year is a leap year, the first day will be 2 index values less than the previous year. Therefore, we can solve for the amount of leap years, multiply this value by 2 and then subtract -1 for the remaining years:

Find leap years from the year 2100 (maximum value):

$$\frac{2100-y}{4} = l$$

The index of the first day of 2100 is 5. In order to find the first day, the first day of 2020 minus the 2 values for leap years and then the 1 values for normal years can be calculated. The third term below is 2100 - y - l is to exclude the leap years that have already been accounted for.

$$5 - 2(l) - (2100 - y - l)\ (mod\ 7) = z$$

Step 2: Find the first day of desired month

Let:

a = increment of V_{ML}
b = increment of V_{MN}
w = index of first day of desired month

The first day of the month can be found using the pattern denoted in table 1. The sum of the V_{MN} V_{ML} can be found respectively for each month to create a table. This table corresponds to the month value with the increments from V_{MN} and V_{MNv}.

Leap year	
$y(mod\ 4) = 0$	
Month	a
January	+0
February	+3
March	+4
April	+7
May	+9
June	+12
July	+14
August	+17
September	+20
October	+22

Non leap year	
$y(mod\ 4) \neq 0$	
Month	b
January	+0
February	+3
March	+3
April	+6
May	+8
June	+11
July	+13
August	+16
September	+19
October	+21

November	+25
December	+27

November	+24
December	+26

The a or b value found from above must be added to z to find the first day of the desired month because adding a or b is the distance to z.

$$(z + a)(mod\ 7) = w$$
or
$$(z + b)(mod\ 7) = w$$

Step 3: Find the day desired

We can use the index for the first day of the desired month as a reference value. The values that are in the same column as the first day will share the same index value as the first day (ie. 8, 15, 22) because all of these values mod 7 will be 1. Therefore, we can find the closest value in the same column as x that is less than the desired day. The same column x value can also be greater than the desired day, ↑both greater than and less than work. The table below demonstrates the idea of reference dates

<table>
<tr><th>Sunday</th><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th><th>Saturday</th></tr>
<tr><td></td><td></td><td></td><td>1</td><td colspan="2" rowspan="3">← distance →</td><td rowspan="2">↑</td></tr>
<tr><td>↑</td><td colspan="2" rowspan="2">← distance →</td><td>8</td></tr>
<tr><td>12</td><td>15</td><td>18</td></tr>
</table>

Equation to find the days in-line with 1:

Find row of the calendar date is in:

$$row = \lfloor d/7 \rfloor$$

Each row increments by 7 as you go down, and the first value is 1, not 0 so:

$$values\ inline\ with\ 1 = \lfloor d/7 \rfloor * 7 + 1$$

With the days in-line with 1 on the calendar, we can find the distance from d to the reference:

$$distance = d - (\lfloor d/7 \rfloor * 7 + 1)$$

The index of the day can be calculated with w plus the distance. However, this value will go beyond the range of the calendar week (0 - 6) so we can do mod 7 to find the index of the day:

$$w + distance\ (mod\ 7) = index\ of\ day$$

Insert the index in the following table to receive the day:

0	1	2	3	4	5	6
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Process 2 (Rianna):
Using the October calendar, we found that the day of Oct 1 of 2022 is a Saturday. We decided to use a 0-index system for the day of the week, since there are 7 days. Monday is 0, Tuesday is 1, Wednesday is 2, all the way until Saturday is 6.

First, we try to find out what day is January 1st, since this is an easier reference date than Oct 1. Let's say that Jan 1st is x, where x is the day of the week in mod 7.
There are 31 days in January, 28 days in February, 31 in March, 30 in April, 31 in May, 30 in June, 31 in July, 31 in August, and 30 in September. We know that October 1st is a Saturday, so the equation we have is

$$x+31+28+31+30+31+30+31+31+31+30 \equiv 6 \pmod{7}$$
$$\Leftrightarrow x+3+0+3+2+3+2+3+3+2 \equiv 6 \pmod{7}$$
$$\Leftrightarrow x+15 \equiv 0 \pmod{7}$$
$$\Leftrightarrow x \equiv -15 \pmod{7}$$
$$\Leftrightarrow x \equiv 6 \pmod{7}.$$

```
$$x+31+28+31+30+31+30+31+31+31+30 \equiv 6 \pmod{7} \\
\Leftrightarrow x+3+0+3+2+3+2+3+3+2 \equiv 6 \pmod{7} \\
\Leftrightarrow x+15 \equiv 0 \pmod{7} \\
\Leftrightarrow x \equiv -15 \pmod{7} \\
\Leftrightarrow x \equiv 6 \pmod{7}.$$
```

So we know that Jan 1st, 2022 is a Saturday.

Now, we can use this as a reference day to find the day of the week of the target date in 2022.
Let's give an example Nov 25th. We add the number of days until Oct 31st, which will be

$$31+28+31+30+31+30+31+31+30+31=304.$$

So Oct 31st is the 304th day of the year. Remember, we need to find Nov 25th, so we simply add 304+25=329th day of the year.
However, just as Jan 2nd is the 2nd day of the year but only 1 day from Jan 1st, Nov 25th is the 329th day of the year but 328 days from Jan 1st. Jan 1st is a Saturday, so we do

$$6 + (329 - 1) \equiv 5 \pmod{7}.$$

Therefore, Nov 25th is a Friday in 2022.

The last step is to find the number of days to switch back. Since

$$365 \equiv 1 \pmod{7}$$

We can simply find the number of years that are in between 2022 and the targeted year. However, for leap years,

$$366 \equiv 2 \pmod{7},$$

So we need to add these extra leap years. The example is 1917.
The closest leap year to 1917 is 1920, and the closest leap year to 2022 is 2020. To find the number of leap years, we do

$$\frac{2020 - 1920}{4} + 1 = 26$$

And we do

$$2022 - 1917 = 105$$

So, the number of days we go back is 105 + 26 = 131 days. Remember, Nov 25th is a Friday, which is 5. Since we are going back in time, we do

$$5 - 131 \equiv -126 \equiv 0 \pmod{7}$$

Therefore, Nov 25th, 1917 is a Sunday. This process can also be extended to future dates.

Process 3 (Vaishnavi/Isabella):
We first found an algorithm to find the day of the week for any date in October 2022 using the given calendar. Using an index from 1-7 (where 1 is Sunday and 7 is Saturday), we found that the remainder of the sum of the date and 6 divided by 7 was the index number.

For example, to find the day of the week of October 24, 2022:

(24 + 6) (mod 7)
30 (mod 7)
2
Monday

After this, we found the formulas for a few more months in a similar manner, allowing us to derive the algorithm to find any day for any date in 2022: (date + (1st day of the month - 1)) mod 7 → day of the week (on a 1-7 index system).

For example, to find the day of the week of June 24, 2022:

The first day is a Sunday, which corresponds to a 1 to our index system.
(24 +(1-1)) (mod 7)
24 (mod 7)

3
Tuesday

Among the given parameters was the minimum of the year 1901 and the maximum of the year 2100 with which our solution should be calculated.

Given 365 days in all non-leap years, dividing it by the number 7 (the amount of days in a week) would give us the number 52 with a remainder of 1. Therefore, every non-leap year, the day of the week for every day moves forward by 1. For example, if January 1st this year was a Saturday, next year it would be a Sunday.
Given 366 days in all leap years, dividing it by the number 7 would give us the number 52 with a remainder of 2. Therefore, a leap year would add two days to the cycle (for all dates after February 28, as the cycle remains the same until the added February 29).

We used January 1, 1901 as a reference date. We calculated the day of January 1st by using a given date: Sunday, August 4, 1901. By finding the number of days between August 4 and January 1 and taking mod 7, we found that January 1 was a Tuesday. Then, we derived an algorithm to find the day of Jan 1st in any given year. To find this algorithm, we underwent the following logical steps.

First, we found the number of years between 1901 and the given date, excluding 1901.
Then, we found the number of leap years by dividing the total number of years by 4, as a leap year occurs every four years.
Using the logic explained earlier, we determined the number of days that Jan 1st would move forward by adding the lower bound (1901) to the number of leap years times 2 and the number of non-leap years, then adding it to 3, which is the index of the first day of 1901, Tuesday. By then finding that number in (mod 7), we can find the first day of any year.

This can be written as the equation below:

$$\left(2\,\mathrm{floor}\left(\frac{(year_f - 1901)}{4}\right) + \left((year_f - 1901) - \mathrm{floor}\left(\frac{(year_f - 1901)}{4}\right)\right) + 3\right) \pmod 7$$

And simplified as:

$$\left((year_f - 1901) + \mathrm{floor}\left(\frac{(year_f - 1901)}{4}\right) + 3\right) \pmod 7$$

To determine the amount of days between January 1st of a particular year and the given date, all of the days of the full months before the date need to be summed together. However, since everything will be eventually manipulated by a mod 7, we can take the mod 7 of each of the sums to reduce calculations. In addition, we made a table of the days of the week for each 1st of the month, assuming January 1st is a Sunday. To use the table to determine the difference in day of the week between January 1st and a particular date in the same year (for every year):

$$[(days\ of\ prior\ months\ +\ day\ of\ Jan\ 1st)\ ]mod7$$

Where the *days of prior months* can be determined from the tables below.

Leap year	
$y(mod\ 4)\ =\ 0$	
Month	Days of prior months
January	1
February	4
March	4
April	7
May	2
June	5
July	7
August	3
September	6
October	1
November	3
December	6

Non leap year	
$y(mod\ 4)\ \neq\ 0$	
Month	Days of prior months
January	1
February	4
March	5
April	1
May	2
June	5
July	7
August	4
September	7
October	2
November	3
December	6

The solution of this expression is a number between 1 and 7, and will return the first day of the target month.

To find the target date, we used the formula explained earlier. We subtracted the first day of the month by 1, and added it to the target date. Then, take the mod 7. The solution to this expression will be a number from 1-7 and will be the final solution.

Solution 1 (Tarun):

Finding the First Day of the Year

1. Calculate (2100 - year) and divide the value by 4. Store the value of the number to the left of the decimal point.
2. Calculate (5 - (2 * the value from step 1) - (2100 - year - stored value from step 1))
    a. If step 2 is negative
        i. Take the positive version of the number and divide by 7, store the remainder
        ii. Calculate 7 - remainder from above
    b. If step 2 is positive
        i. Take the value from step 2 and divide by 7, store the remainder

Finding the First Day of the Month

3. Using the values from clause 2 of 2a or clause 1 of 2b:
    a. If the month is a leap year (the last two digits are divisible by 4)
        i. Add the corresponding value to your month from this table to the value from clause 2 of 2a or clause 1 of 2b
        ii. Take the value from above, divide by 7, and store the remainder
    b. If the month is not a leap year (the last two digits are not divisible by 4)
        i. Add the corresponding value to your month from this table to the value from clause 2 of 2a or clause 1 of 2b
        ii. Take the value from above, divide by 7, and store the remainder

Finding the Day of the Month

4. Divide the day (that you want) by 7 and using the value left of the decimal point, multiply by 7 and then add 1
5. Calculate (value of the day (that you want) - the value from step 4)
6. Take the value from clause 2 of 3a or clause 2 of 3b (whichever was calculated) and add the value from step 5
    a. If the value from step 6 is negative
        i. Take the positive version of the number and divide by 7, store the remainder
        ii. Calculate 7 - remainder from above
    b. If the value from step 6 is positive
        i. Take the value from step 6 and divide by 7, store the remainder
7. The value from clause 2 of 7a or clause 1 of 7b will correspond to a day in this table. That is the day for your date.

Solution 2 (Rianna):

1. Calculate the number of days per month until the targeted date. (Ex: Say you're doing August, add up 31 (January) + 28 (February) + 31 (March) + 30 (April) + 31 (May) + 30 (June) + 31 (July).)
2. Then add the date of the targeted date to what you got in part 1. (Ex: August 15th - add 15 to what you got in part a.)
3. Divide this number by 7 and save the remainder.
4. Now, subtract the year in the given date and subtract it from the current year, and save this number. (Ex: 2022 - 1989).
5. Go to step 6 if the given date is a leap year, step 7 if the given date is a non-leap year.
6. Subtract the given date from 2020, and divide by 4. (Ex: if the given date is 1988, find (2020-1988)/4+1.)
    a. Add this to what you got in part 4, and take the mod.
    b. Add this to what you got in part 3, and take the mod.
    c. This will be your answer.
7. Find the closest leap year to the given date. (Ex: the closest leap year is 1989 to 1992).
    a. Subtract this leap year from 2020, divide by 4, and add 1. (Ex: (2020 - 1992)/4 + 1 = 8)
    b. Add this to what you got in part 4, and take the mod.
    c. Add this to what you got in part 3, and take the mod.
    d. This will be your answer.

Solution 3 (Vaishnavi/Isabella):

1. Find the day of January first in the target year.
    a. Subtract the target year from 1901. This is the total number of years.
    b. Find the number of total years divided by 4, and round down (floor). This is the amount of leap years.
    c. Add the two values from a) and b).
    d. Add this to 3, which is the index of the first day of 1901.
    e. Find the remainder of this value when divided by 7 (that is, take mod 7).
2. Find the day of the target date.
    a. Using the table, find the amount that should be added to the value found in step 1 for the target month. (add one less than the table value) Then take the mod 7 of this new value. This is the first day of the target month.
    b. Subtract this value from 1 and add to the target date. Then take the mod 7
    c. This will be your final answer.

Extensions:

The aforementioned methods could be further generalized such that they are compatible with any date, despite its year. Currently, the methods would produce inaccurate results if the target date was before 1900 or after 2100 because the calculations cannot account for the fact that there was not a leap year in those years despite being multiples of 4.

Additionally, the methods could be reversed to determine the date of certain holidays in a specific year. For example, a method could be derived to calculate the date of Thanksgiving or Labor Day in a given year.

Another possibility is to adapt these methods to a Lunar Calendar. A problem could involve finding the corresponding Lunar date of a given Gregorian one, or vice versa. Using that method, one could produce a method to determine the Gregorian date that Hanukkah would fall on in a particular year.