Mihika Chalasani, Anyee Li, Anne Wu, Amy Chen

POW #1: Happy Birthday

Problem Statement:

How can you determine what day of the week you were born on given only a calendar of the current month and how many days are in each month?

Process:

- Anyee's birthday happened to be in October (the given calendar month) and she knew what day she was born on, so we tested everything with her birthday first.
- First, we realized that there are 365 days in a year and, thus, 52 weeks and one day. This one day remainder means that if your birthday is on a specific day of the week this year, last year it was the previous day of the week and next year it will be the next day of the week (not including leap years)
    - For example, if your birthday is on Monday this year, it was on Sunday last year and will be on Tuesday next year (again, not including leap years)
- We then tried to see how leap years would affect this pattern:
    - We realized that if your birthday is after the leap day on leap year, it gets shifted two days forward that year (compared to the previous year)
    - If your birthday was before the leap day, it gets shifted two days forward in the year after the leap year.
- We then spent some time trying to figure out how to make a formula to account for birthdays shifting on and after the leap year
- After none of this brainstorming lead to any breakthroughs regarding leap years, we decided to ignore the differences in birthdays before and after leap years until we found a counterexample
- Looking at Anyee's birthday, we found that she is turning 17 this year, has lived through 4 leap years, and that her birthday this year is the same day of the week as when she was born.
- From this, we hypothesized that perhaps every 17 years, your birthday returns to the original day it was when you were born
    - Thus, our solution would have been to find your next birthday in which you turn a multiple of 17 and determine what day of the week that is
    - However, this did not work for every birthday- it seems it only worked for this specific case because when we added the age and the number of leap years, it came out to a multiple of 7
    - This theory does not hold up when the age added with the number of leap years isn't a multiple of 7
- We started to look at mods and realized that if Anyee's birthday went forward one day each year for 17 years (because she is turning 17), plus an extra four days for the four leap years, you get 21 days forward, which in mod 7, is 0. This means that her birthday was on the same day of the week in 2005 as it is in 2022.

- Since we had a vague outline of our model, we decided to figure out how to determine what day a person's birthday would be on this year.
    - We concluded that if you figure out the number of days between a basis day (usually today), and your birthday, and mod 7 that number, the answer would be the number of days you would have to move forward or backwards to get the day of the week of your birthday this year
        - After some trial and error, we figured out that if you are determining the day of the week for a past date, you would go backwards. However, if the date is in the future, you go forwards
- After figuring out how to find the day of the week of a certain date, we looked at determining how many leap years a person has lived through.
    - Initially, we believed that the person's age divided by 4 and ignoring the remainder would get us the number of leap years.
    - However, this was disproven as a counterexample was found where a 15 year old person had lived through four leap years
    - Anne had developed a rough formula to calculate the number of leap years, and after a few iterations and trial and error, we solidified that to easily calculate the number of leap years.
- After writing down our method, testing out multiple dates, and having others test our methods, we determined that our model worked

Solution:

We know our solution is complete because after testing it for multiple different dates, it worked for all of them. In these tests, we tested a birthday that was before the leap day, a birthday that was after the leap day, a birthday the year before a leap year, and a birthday on a leap year. Since we tested a wide variety of birthdays, including the dates that would be most likely to cause a problem in our model (ones including leap years), we can conjecture that our model is concrete enough for birthdays between 1901 and 2022.

1. Determine what day of the week today is
2. Find the number of days between your birthday this year and today

    Here is a table of the number of days in each month:

Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec
31	28*	31	30	31	30	31	31	30	31	30	31

    *29 in leap years

    a. Note if your birthday has already passed or if it is upcoming
3. Divide that number by 7 and find the remainder (mod)
4. If your birthday was in the past, go back however many days the remainder came out to be. If your birthday is upcoming, go forward instead (from today's day of the week)!
    a. This is the day of the week your birthday was or is going to be this year
5. Next, find the number of leap years you have lived through (if a year is divisible by 4, it is a leap year)
    a. Find the nearest future leap year from the current year

       b. Find nearest future leap year after your birth year
            i. If you were born on a leap year before Feb 29, count the current year as the nearest leap year
            ii. If you were born on a leap year after Feb 29, count the next leap year as the nearest leap year
       c. Subtract the year from step b from the year from step a
       d. Divide this number by 4 to find the number of leap years you have lived through
6. Add this number to your age at your birthday this year. This means that if your birthday is still upcoming, take the age you will be on your birthday
7. Divide this number by 7 and find the remainder
8. Go backwards that number of days from the day of the week your birthday is or was this year
9. This day of the week is the day of the week you were born!

Extensions:

Our model currently only works to determine birthdays from 1901 to 2022. However, it could be easily extended to calculate birthdays in the future up to 2099 as well.

Steps 1-4 would be the same:
1. Determine what day of the week today is
2. Find the number of days between your birthday this year and today

Here is a table of the number of days in each month:

Jan	Feb	Mar	Apr	May	June	July	Aug	Sept	Oct	Nov	Dec
31	28*	31	30	31	30	31	31	30	31	30	31

*29 in leap years

       a. Note if your birthday has already passed or if it is upcoming
3. Divide that number by 7 and find the remainder (mod)
4. If your birthday was in the past, go back however many days the remainder came out to be. If your birthday is upcoming, go forward instead (from today's day of the week)!
       b. This is the day of the week your birthday was or is going to be this year

Step 5 would be slightly modified:
5. Instead of finding the number of leap years you have lived through, you can calculate the number of leap years between this year and the target year (year you want to find your birthday on)
       a. Find the nearest future leap year from the current year
       b. Find nearest future leap year <u>to your target year</u>
            i. <u>If this year is a leap year and your birthday is before</u> Feb 29, count the current year as the nearest leap year
            ii. <u>If this year is a leap year and your birthday is after</u> Feb 29, count the next leap year as the nearest leap year
       c. Subtract the year from step b from the year from step a

          d. Divide this number by 4 to find the number of leap years you have lived
through

6. Instead of using your age, you can take the number of years between this year and your target year and add the number of leap years from the previous step
7. Divide this number by 7 and find the remainder
8. Go <u>forwards </u>that number of days from the day of the week your birthday is or was this year
9. This day of the week is the day of the week your birthday will be on that year!

We also created a Java implementation of our original model. It is shown below and works for all cases the original model could handle as well. (This means it doesn't include the future dates extension)

```java
import java.util.Scanner;
public class POWBirthday {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String[] daysOfTheWeek = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};
        int[] daysInMonth = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

        System.out.println("Enter the day you were born: ");
        int day = scanner.nextInt();
        System.out.println("Enter the month you were born (as a number): ");
        int month = scanner.nextInt();
        System.out.println("Enter the year you were born: ");
        int year = scanner.nextInt();

        int knownDay = 7;
        int knownMonth = 10;
        int knownYear = 2022;
        int knownDOW = 5;


        int daysBetween = -1;
        boolean fwdOrBwd = true; // if true, going forwards, if false, going
backwards
        if (knownMonth > month) {
            fwdOrBwd = false;
            daysBetween = daysInMonth[month] - day;
            for (int i = month + 1; i < knownMonth; i++) {
                daysBetween += daysInMonth[i];
            }
            daysBetween += knownDay;
        }
        else if (knownMonth < month) {
            fwdOrBwd = true;
            daysBetween = day;
            for (int i = month - 1; i > knownMonth; i--) {
                daysBetween += daysInMonth[i];
            }
```

```java
                daysBetween += daysInMonth[knownMonth] - knownDay;
        }
        else if (knownMonth == month) {
                if (knownDay > day) {
                        fwdOrBwd = false;
                        daysBetween = knownDay - day;
                }
                else if (knownDay < day) {
                        fwdOrBwd = true;
                        daysBetween = day - knownDay;
                }
                else if (day == knownDay) {
                        fwdOrBwd = true; //defaulting
                        daysBetween = 0;
                }
        }

        int rangeForDOW = 0;
        if (fwdOrBwd) {
                rangeForDOW = knownDOW + daysBetween % 7;
        }
        else {
                rangeForDOW = knownDOW - daysBetween % 7;
        }
        if (rangeForDOW < 0) {
                rangeForDOW += 7;
        }
        else if (rangeForDOW > 7) {
                rangeForDOW -= 7;
        }

        int currentFutureLeap = knownYear;
        for (; currentFutureLeap % 4 != 0; currentFutureLeap++) {}

        int pastFutureLeap = year;
        if (month <= 2 && pastFutureLeap % 4 == 0) {}
        else if (pastFutureLeap % 4 == 0) {
                pastFutureLeap += 4;
        }
        else {
                for (; pastFutureLeap % 4 != 0; pastFutureLeap++) {}
        }

        int numofLeapDays = (currentFutureLeap - pastFutureLeap) / 4;

        int age = knownYear - year;

        rangeForDOW = (rangeForDOW - age - numofLeapDays);
        while (rangeForDOW < 0) {
                rangeForDOW += 7;
        }

        System.out.println(daysOfTheWeek[rangeForDOW]);
    }
```

Mihika Chalasani, Anyee Li, Anne Wu, Amy Chen

```
}
```

Sample sequence of inputs:

```
Enter the day you were born:
12
Enter the month you were born (as a number):
9
Enter the year you were born:
2006
Tuesday
```