

1. Problem statement

Have you ever wondered if your birthday fell on the same day of the week as Michelle Obama? Fortunately for you, the problem presented to us was developing a mathematical approach to determine the day of the week one was born on, with our only resources being the current month's calendar and a sample list of famous individuals' birthdays with the days of the week they fell on.

2. Process

The first thing that stood out to us was the fact that every 4 years, it is a leap year with one extra day added to February. We wanted to eliminate this problem as soon as possible, so we started by finding the sum of the days in a 4 year cycle. Specifically, this consisted of adding $365 + 365 + 365 + 366$. The sum was 1461. We modded this number by 7 to find the shift in days of the week every 4 years. The remainder was 5, hence indicating that there was a shift of 5 days every 4 years. So, for every "normal" year (365 days), there was a 1 day shift ($365 \bmod 7$).

Next, we decided to find the day of the week that January 1, 1901 fell on, so we could calculate the days between our target date and 01/01/1901, then calculate the number of shifts in days of the week. The closest day we could find to that date was August 4th, 1901, Louis Armstrong's birthday, which was on a Sunday. We first identified that August 1st was a Thursday (4 days before), then with the help of the table showing the number of days in each month, we found that there were 212 days between January 1st 1901 (inclusive) and August 1st (exclusive). We modded 212 by 7 to find the number of shifts in days of the week again, and we got 2. As a result, we jumped back 2 days from Tuesday — since we were going backwards in time — and came to the conclusion that January 1st 1901 was on a Tuesday.

Therefore, in the last step of our method, we always make the number of shifts starting from Tuesday, as all the previous steps of the process are solving for the difference in days from the target date and 01/01/1901, then modding it by 7 to find the number of shifts. We would like to mention here that in the early stages of our problem-solving process, we were considering finding the day of the week of January 1, 2000, since it was around the middle of the dates, but we decided against it to avoid confusion with addition vs subtraction based on the target date being before or after 2000.

Even after identifying the overall structure of our method, we ran through one significant issue that made us question our initial steps. When we were testing our strategy with Anwar Sadat's birthday (December 25, 1918), we found the number of extra days that could not be made into a full week (modding 7) between 01/01/1901 and 12/25/1918, but when we performed that number of jumps forward starting from Tuesday, we ended up one day after the actual day it should've been. This occurred again when we attempted with another given date, which led us to think that we made an error finding the day of the week for January 1st 1901.

However, we realized that we actually had to subtract one from the difference in number of days as we were double counting the 1st day of the year.

To compare this with our actual correct method, we split the period of time between 01/01/1901 and the target date into 2 chunks: the years between 1901 and the year right before the year of the target date (both inclusive, so technically 01/01/1901 - 12/31/xxxx-1), and the number of days in the year of target date, including the target date itself. In our earlier approach, we had counted January 1st of the target date year twice, so to account for this, after we found the sum of remaining days for the years (5 extra days * every leap year cycle, 1 extra day * every extra year), we subtracted one before adding it with the sum of the remaining (after modding 7) days from the actual year itself.

After several iterations, we came up with our final method, and the specific steps can be seen (and followed!) below. We hope you follow along and find out if you really were born on the same day of the week as Michelle Obama!

3. Solution

- a. Identify the target date (the date you want to find the day of the week for).
 - i. It should be between 1901 and 2099 (inclusive).
- b. Subtract 1901 from your birth year.
- c. Divide the difference in years by 4. Take note of the quotient and remainder.
- d. Multiply the quotient by 5, then add to the remainder.
- e. Subtract one from the previous sum.
- f. Divide this new sum (with one subtracted) by 7 and find the remainder. Let this remainder be r1.
- g. Add up the dates leading up to the date of the target date.

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

- i. Specifically, find the sum of the number of days of each month leading up to the target date, then add the specific day of the target date to the sum.
- h. Divide this sum by 7 and find the remainder. Let this remainder be r2.
- i. Sum r1 (from step f) and r2 (from step h).
- j. Divide this number by 7 again and find the remainder.
- k. Starting from Tuesday at index 0, Wednesday at index 1, Thursday at index 2, etc., find the corresponding day of the week based on the new remainder.

Tues	Wed	Thur	Fri	Sat	Sun	Mon
0	1	2	3	4	5	6

- i. This is the day of the week of the target date. Congratulations!

4. Extensions

- a. Additional question / slightly different aspect of the problem -> How can we accomplish a similar goal of determining the day of the week one was born on — with the same resources of the current month's calendar and a sample list of famous birthdays — if they were born in a century year that is not a leap year (not divisible by 400) such as 1800 and 1900? What would the new method be for any year, knowing that every 4th century year is a leap year (while the others are not)?
- b. Code for finding day of the week - submitted separately to the POW #1 assignment

```
Problems Javadoc Declaration Console X
<terminated>- BirthdayPOW [Java Application] /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java (Oct 13, 2023, 10:53:24 PM - 10:54:08 PM) [pid: 19931]
Please enter the name of your birthday month.
February
Please enter your the day of the month your birthday is on.
4
Please enter the full year of your birthday (XXXX).
2078
|
This birthday was on a Friday.
```

```
Problems Javadoc Declaration Console X
<terminated>- BirthdayPOW [Java Application] /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java (Oct 13, 2023, 10:51:41 PM - 10:51:57 PM) [pid: 19880]
Please enter the name of your birthday month.
May
Please enter your the day of the month your birthday is on.
3
Please enter the full year of your birthday (XXXX).
1956
|
This birthday was on a Thursday.
```

```
Problems Javadoc Declaration Console X
<terminated>- BirthdayPOW [Java Application] /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home/bin/java (Oct 13, 2023, 10:55:35 PM - 10:55:55 PM) [pid: 19969]
Please enter the name of your birthday month.
October
Please enter your the day of the month your birthday is on.
30
Please enter the full year of your birthday (XXXX).
1986
|
This birthday was on a Thursday.
```

```
1 import java.util.Scanner;
2
3 public class BirthdayPOW {
4
5     public static void main String[] args) {
6
7         Scanner pow1 = new Scanner(System.in);
8         System.out.println "Please enter the name of your birthday
9 month.");
10        String month = pow1.nextLine();
11        int a = 0;
12        switch (month.toLowerCase()) {
13            case "january":
14                a = 1;
15                break;
16            case "february":
17                a = 2;
18                break;
19            case "march":
20                a = 3;
21                break;
22            case "april":
23                a = 4;
24                break;
25            case "may":
26                a = 5;
27                break;
28            case "june":
29                a = 6;
30                break;
31            case "july":
32                a = 7;
33                break;
34            case "august":
35                a = 8;
36                break;
37            case "september":
38                a = 9;
39                break;
40            case "october":
41                a = 10;
42                break;
43            case "november":
44                a = 11;
45                break;
```

```
45     case "december":
46         a = 12;
47         break;
48     default:
49         a = -1;
50     }
51
52     Scanner pow2 = new Scanner(System.in);
53     System.out.println("Please enter your the day of the month
your birthday is on.");
54     String day = pow2.nextLine();
55     int b = Integer.parseInt(day);
56
57     Scanner pow3 = new Scanner(System.in);
58     System.out.println("Please enter the full year of your
birthday (XXXX).");
59     String year = pow3.nextLine();
60     int c = Integer.parseInt(year);
61
62
63     int q = ((c-1901)/4)*5-1;
64     int r = (c-1901)%4;
65     int m = 0;
66
67     if(r == 3) {
68         switch (a) {
69             case 1:
70                 m = 0;
71                 break;
72             case 2:
73                 m = 31;
74                 break;
75             case 3:
76                 m = 60;
77                 break;
78             case 4:
79                 m = 91;
80                 break;
81             case 5:
82                 m = 121;
83                 break;
84             case 6:
85                 m = 152;
86                 break;
87             case 7:
```

```
88         m = 182;
89         break;
90     case 8:
91         m = 213;
92         break;
93     case 9:
94         m = 244;
95         break;
96     case 10:
97         m = 274;
98         break;
99     case 11:
100        m = 305;
101        break;
102    case 12:
103        m = 335;
104        break;
105    default:
106        m = -1;
107    }
108 }
109 else {
110     switch (a) {
111     case 1:
112         m = 0;
113         break;
114     case 2:
115         m = 31;
116         break;
117     case 3:
118         m = 59;
119         break;
120     case 4:
121         m = 90;
122         break;
123     case 5:
124         m = 120;
125         break;
126     case 6:
127         m = 151;
128         break;
129     case 7:
130         m = 181;
131         break;
132     case 8:
```

```
133         m = 212;
134         break;
135     case 9:
136         m = 243;
137         break;
138     case 10:
139         m = 273;
140         break;
141     case 11:
142         m = 304;
143         break;
144     case 12:
145         m = 334;
146         break;
147     default:
148         m = -1;
149     }
150 }
151
152 int dD = (q+r+m+b)%7;
153
154 String finDate = new String();
155
156 switch (dD) {
157     case 0:
158         finDate = "Tuesday";
159         break;
160     case 1:
161         finDate = "Wednesday";
162         break;
163     case 2:
164         finDate = "Thursday";
165         break;
166     case 3:
167         finDate = "Friday";
168         break;
169     case 4:
170         finDate = "Saturday";
171         break;
172     case 5:
173         finDate = "Sunday";
174         break;
175     case 6:
176         finDate = "Monday";
177         break;
```

BirthdayPOW.java

Friday, October 13, 2023, 10:42 PM

```
178     }
179
180     System.out.println "\n" + "This birthday was on a " +
    finDate + ".";
181 }
182 }
183
```