Lily, Marlon, Kweku

# Birthday POW #1 Write Up

1. Problem Statement:

Many individuals in the world know the date they were born, and in fact, often have "birthday parties" or celebrations to commemorate this. However, one fault of this knowledge is that such individuals lack the knowledge of what specific day they were born on. Some individuals, such as one student at MAMS, named Kweku, knows the day he was born due to the fact that his name indicates it ("Kweku" means born on Wednesday). However not everyone has such a resource. Our task was to create an algorithm or method that allows anyone to find the day of the week they were born on in a simple manner.

2. Process:

First, We started off by labeling the days of the week 0-6, starting on Sunday. This means that Sunday is Day 0, Monday is 1, Tuesday is 2, and so on. Then, we saw from the given calendar that October first is a Saturday, or 6. We used this knowledge and the days in each month to find that January first, 2023, is on a Sunday. To do this, we did (7+31+30+31) mod 7 = 99 mod 7 = 1, which is Sunday.

Initially, our plan was to find a way to determine what January first is for different years, which would allow us to use the year a person was born and add on the number of days per month until the month they were born in and the days of the month they were born in until their birthday. After finding 1/1/2023, we added 365 days mod 7 to find that 1/1/2024 is on a Monday. This showed that for every normal year, the date from one year to the next is the weekday +1. Because 2024 is a leap year, we added 366 days to find that 1/1/2025 is on a Wednesday. This showed us that for every leap year, the date from one year to the next is the weekday +2. With this knowledge, we changed our model to be based on the current date, not January first. Link to the first model:

https://docs.google.com/drawings/d/12ZdtTK4Ae8yUeTkpVpcEouAFIqzxbcj8RYK2BW0KR1M

After creating a model for the current date, we created a second, simpler, new model based again on January first that is more accurate. Link to doc with photos of the process:

https://docs.google.com/document/d/1WSMEY7vrll55a9_0eE0NcQB5mqJ7SYT0C9bukGjkoqo

3. Solution:

Link to Final Model:

https://docs.google.com/drawings/d/1c7W2Vy6rSZ_EmZXHtZ0E4R15psu6XT5953T5p2IlyQI

Explanation of Completion and Accuracy:

After creating our initial model, we tested it with a set of different birthdays, most of which were successful. However, we discovered that the initial model didn't work for all birthdays. To correct this error, we created a second model that, when tested, was successful. Based on our tests, we can infer that our new model should be accurate for all birthdays between 1900 and 2100.

4. Extensions:

- Finding Birthdays before Common Era (BC, not AD)
- Birthdays before 1900
    - Leap year every four centuries
- Given length of life in hours, find time of day they were born or vice versa
- Separate calculation for the future birthday
- Create a computer program to calculate this