Programming #5C: A better birthday validation program

We want to redo the leap year and date validation task in Program #3 by defining and calling functions for date validation and checking leap years. Compared with what you did in Program #3, it should make the program much more compact and easy to understand. See the suggested layout of your program below.

Layout of your program:

The following is the suggested layout of your program:

```
#include <iostream>
using namespace std;

//Define the isLeapYear function in the following.
//It should return true if year represents a leap year;
//otherwise, it should return false.
bool isLeapYear(int year)
{      //Fill in the details of your code for the isLeapYear function
}

//Define the isValidDate function in the following
//It should return true
//if month, day, and year together represents a valid date;
//otherwise, it should return false.
bool isValidDate(int month, int day, int year)
{      //Fill in the details of your code for the isPrime function
}

//Define the main function in the following
int main()
{      //Fill in the details of your code for the main function
}
```

What you should do in each function:

1. *bool isLeapYear(int year)* : You should implement the *isLeapYear*(int year) function such that it will return the Boolean value *true* whenever it is called with an argument that is a leap year and otherwise will return *false*. **Do not** ask for user input in this function. **Do not** print out any message in this function.

2. ***bool isValidDate(int month, int day, int year)*** : You should implement the *isValidDate(int month, int day, int year)* function such that it will return the Boolean value *true* whenever the numbers (*day, month,* and *year)* compose a valid date and otherwise it will return *false*. In the implementation, you should call the ***isLeapYear*** function whenever you need to determine whether the year is a leap year to determine whether it is a valid date in February. **Do not** ask for user input in this function. **Do not** print out any message in this function.

3. *int main()* : Implement the main function to do the following things:

- **Read and verify the user's birthday entered to ensure it is a valid date**: In the main function, ask the user to provide the information of the birthday (the day, the month, and the year of each birthday as three integers respectively) of himself/herself. **For the input of each birthday, use a loop to repeatedly check whether the three numbers entered by the user (i.e. the day, the month, and the year of the birthday) compose a valid date by calling the *isValidDate* function.** If the numbers (the day, the month, and the year of the person's birthday) given do not compose a valid date, report an "Invlaid Date" message and ask the user to input the numbers. **The loop should repeat until the three numbers (the day, the month, and the year of the person's birthday) compose a valid date.**
- **Report whether the user was born in a leap year**: Determine and report whether the user's birthday is in a leap year or not. You should call *isLeapYear* function and check whether the year is a leap year by using an *if* statement to check whether the return value is *true* or not.

- **Do the two things above for the father's birthday and then for the mother's birthday too**.

About leap years:

A year is a leap year if it is divisible by 4 **except that** any year divisible by 100 is a leap year **only if** it is also divisible by 400. So 1900 is not a leap year, but 2000 is, i.e.

- a year (e.g. 1996) is a leap year if it is divisible by 4 but not by 100,

- a year (e.g. 2000) is a leap year if it is divisible by 400 (and thus by 100 too),
- otherwise, it is not a leap year.

About valid dates: three integers *day*, *month*, and *year* compose a valid date if and only if *year* is positive and one of the following cases is true

- *month* is one of 1, 3, 5, 7, 8, 10, or 12, and (ii) *day* is in the range of [1, 31],
- *month* is one of 4, 6, 9, or 11, and (ii) *day* is in the range of [1, 30],
- *month* is 2 and *year* is a leap year, and (ii) *day* is in the range of [1, 29], and
- *month* is 2 and *year* is not a leap year, and (ii) *day* is in the range of [1, 28].