

Programming Assignment #3: A Date Validation Program

Task:

1. **Get the birthday information:** Ask the user to provide the information of his/her birthdays (the day, the month, and the year of the birthday as three positive integers respectively). You should declare three separate *int* variables for the person to store the day, the month, and the year of the birthday information.
2. **Check and report the leap year information:** We accept any positive integer as a valid year. If the year is a positive number, determine and report to the user whether that person was born in a leap year or not, otherwise report to the user that the year is not a positive year for that particular person.
3. **Check and report the validity of the date:** Check whether the birthday information is valid. If the birthday information is invalid, report to the user that the year is not valid. For example, no one could be born on 2/29/1982 or 9/31/1985 or 12/0/1990 or 14/5/1995. None of them is a valid date.
4. **Repeat Steps 1 to 3 above** to get and check the birthday information of the user's father and the user's mother respectively. For each parent, you should declare three separate *int* variables for the person to store the day, the month, and the year of the birthday information.
5. **Check and report the consistency of dates:** Check to make sure the user is younger than both his / her father and mother. If this is not the case, report the finding to the user and tell the user that he / she should not be older than the parents.

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. In other words,

- 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,
- otherwise, it is not a leap year.

Comparing dates:

- To compare two birthdays, you should compare the years first, then the months if there is a tie, and then the days if again you have a tie.
- Alternatively, for each date you can get a “*birthday integer*” by multiplying year by 10000, multiplying the month by 100, multiplying the day by 1, and then add these three numbers together. Then you can compare two dates by comparing their corresponding *birthday integers*.