BMCC – Fall 2021
Advanced Programming Techniques

Date Class Project

In this project, you should design a **Date** class and use it to create a calendar application that has many features. You are not limited to the features included in these instructions, feel free to add any other components you find useful.

Note: A normal year has 365 days. A leap year has 366 days (the extra day is February 29). In this project, you must take this fact into account. The rules that determine leap years are provided later in these instructions.

Design the **Date** class, to be used in your application, containing:

- A private data member **month** of type integer that holds the date's month.
- A private data member **day** of type integer that holds the date's day.
- A private data member **year** of type integer that holds the date's year.
- A default constructor that sets the date to January 1, 1753.
- Accessors for each member variable.
- Mutators for each member variable.
- A member function that prints on the screen a date in the form *mm / dd / yyyy*.
- A member function that returns the day name of a date (ex, Sunday, Monday …). Note that the day for January 1 of the year 1753 was a Monday.
- A member function that prints a date including the name of the day and the name of the month as strings (ex: Tuesday, April 6, 2021).
- The overloaded subtraction operator (–) that returns an integer that is the difference in days between two **Date** objects.
- The overloaded increment operator (++) that modifies the **Date** object so that it represents the next day.

- The overloaded decrement operator (`--`) that modifies the **Date** object so that it represents the previous day. The operator must not decrement dates prior to January 1, 1753.

Write a program that thoroughly tests your **Date** class. Consider using the following functions in your program (you can use more functions). The implementations of these functions are left to you; they can be members of the **Date** class, friends of it, or just regular stand-alone functions.

- A function that tests whether a year is leap.
- A function that calculates and returns a date by adding a fixed number of days to the current date.
- A function that calculates and returns a date by subtracting a fixed number of days from the current date.
- A function that returns the number of days passed in the current year.
- A function that returns the number of days remaining in the current year.
- A function that prints the calendar for the current month.
- A function that prints the holidays of any year.

Rules for determining leap years:

✓ Leap Years are any year that can be exactly divided by 4 (such as 2016, 2020, 2024, etc)
  ✗ Except if it can be exactly divided by 100, then it isn't (such as 2100, 2200, 2300, etc)
    ✓ Except if it can be exactly divided by 400, then it is (such as 2000, 2400, 2800, etc)

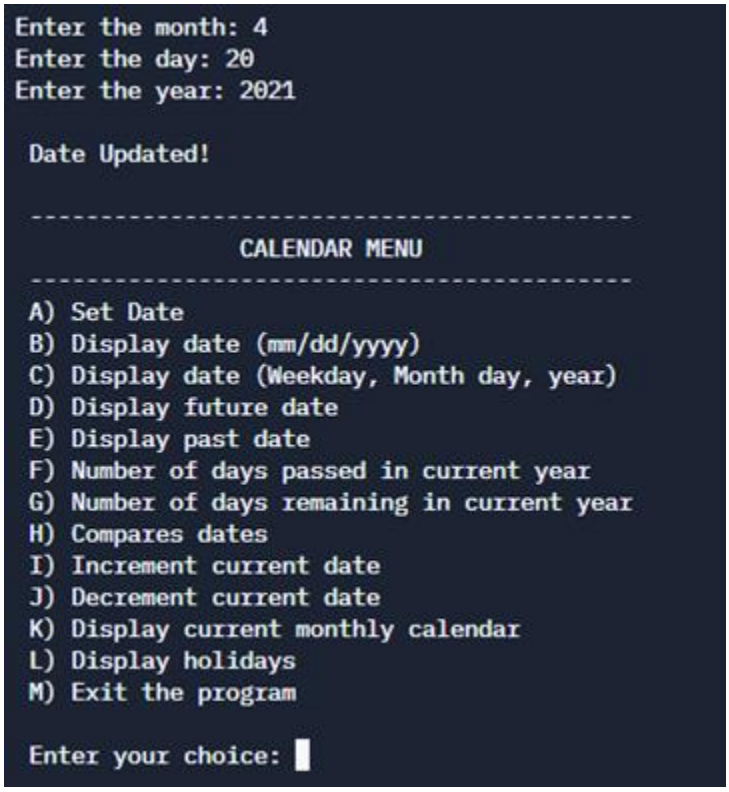
List of US federal holidays:

- New Year's Day - January 1st
- Martin Luther King, Jr. Day - Third Monday of January
- President's Day - Third Monday of February
- Memorial Day - Last Monday of May
- Independence Day - July 4th
- Labor Day - First Monday of September
- Columbus Day - Second Monday of October
- Veterans Day - November 11th
- Thanksgiving Day - Fourth Thursday of November
- Christmas Day - December 25th

Sample application demonstration:

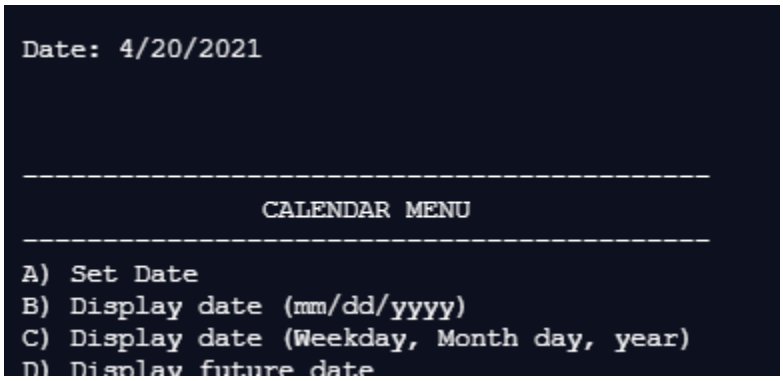
Main menu

Choice A: Set date

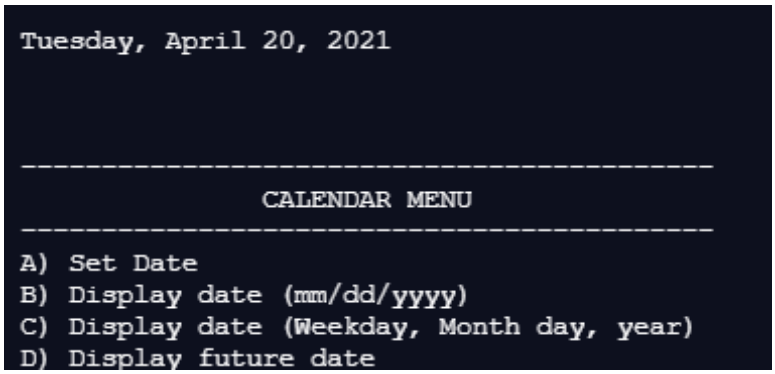


Date should not be set if the entered date is not valid (ex, February 31, 2020).

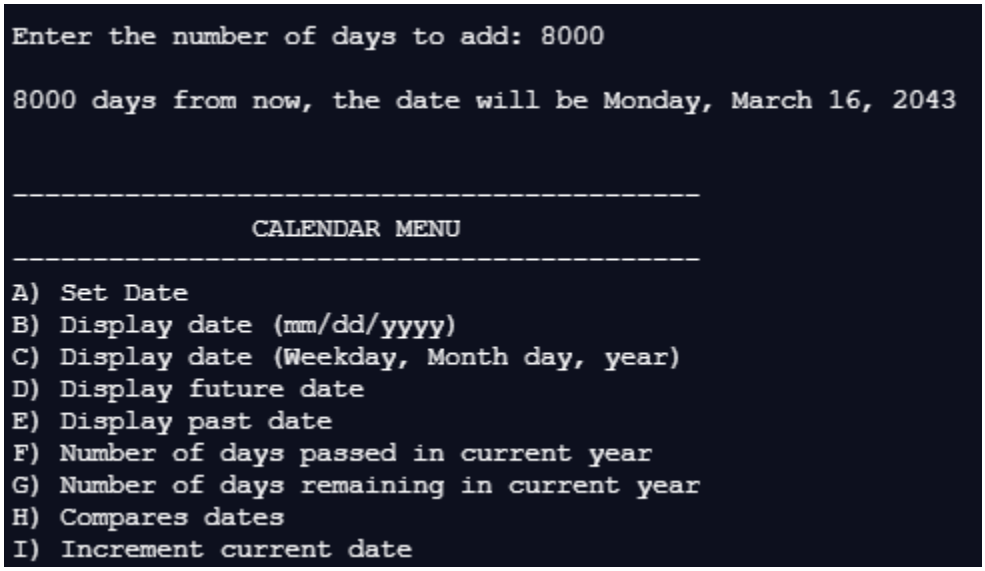
Choice B: Display date (mm/dd/yyyy)



Choice C: Display date (Weekday, Month day, year)



Choice D: Display future date



Choice E: Display past date

```
Enter the number of days to subtract: 200

200 days ago, the date was Friday, October 2, 2020


-----------------------------------------
               CALENDAR MENU
-----------------------------------------
A) Set Date
B) Display date (mm/dd/yyyy)
C) Display date (Weekday, Month day, year)
D) Display future date
E) Display past date
F) Number of days passed in current year
G) Number of days remaining in current year
```

Choice F: Number of days passed in current year

```
109 days have passed this year.



-----------------------------------------
               CALENDAR MENU
-----------------------------------------
A) Set Date
B) Display date (mm/dd/yyyy)
C) Display date (Weekday, Month day, year)
D) Display future date
E) Display past date
F) Number of days passed in current year
G) Number of days remaining in current year
```

Choice G: Number of days remaining in current year

```
255 days left this year.



-----------------------------------------
               CALENDAR MENU
-----------------------------------------
A) Set Date
B) Display date (mm/dd/yyyy)
C) Display date (Weekday, Month day, year)
D) Display future date
E) Display past date
```

Choice H: Compare dates

```
Enter the month: 2
Enter the day: 10
Enter the year: 2022

The date entered will be reach in 296 days!
```

Choice H (again): Compare dates

```
Enter the month: 3
Enter the day: 11
Enter the year: 2021

The date entered has passed 40 days ago!
```

Choice I: Increment current date

```
Date incremented!

Wednesday, April 21, 2021
```

Choice J: Decrement current date

```
Date decremented!

Tuesday, April 20, 2021
```

Choice K: Display current monthly calendar

```
      April 2021
 Su Mo Tu We Th Fr Sa
              1  2  3
  4  5  6  7  8  9 10
 11 12 13 14 15 16 17
 18 19 20 21 22 23 24
 25 26 27 28 29 30
```

Choice L: Display holidays

```
Enter the year: 2021

Friday, January 1       - New Year's Day
Monday, January 18      - Martin Luther King, Jr. Day
Monday, February 15     - President's Day
Monday, May 31          - Memorial Day
Sunday, July 4          - Independence Day
Monday, September 6     - Labor Day
Monday, October 11      - Columbus Day
Thursday, November 11   - Veterans Day
Thursday, November 25   - Thanksgiving Day
Saturday, December 25   - Christmas Day
```

Linux calendar command:

To check the results of your application, you can use the Linux **cal** command which displays the calendar of a specific month or an entire year as follows.

- **cal** without any arguments will display the calendar for the current month.
- **cal** with year as argument will display the calendar for the whole year (ex, `cal 2022`).
- **cal** with month and year as arguments will display the calendar for that specific mount (ex, `cal 12 2021`).

If you don't have access to a Linux terminal, you can use any online Linux emulator such as the following: https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192