CSCI 121 Project 05

Introduction

Purpose

This project will help the students to reinforce the knowledge from Chapter 5 of the textbook.

Objectives

1. Understand and apply top-down design concept
2. Implement the functions according to specified requirements
3. Test and debug the functions

Please read the project 7 and 8 (Two projects about calendar) on pages **301** to **303** of textbook. Your Project 5 will implement and test all the functions listed in these two projects. These functions are:

```
void testMenu();
// post-condition: the test menu is displayed for choose

bool isLeapYear(int year);

int getCenturyValue(int year);

int getYearValue(int year);

int getMonthValue(int month, int year);

int dayOfWeek(int month, int day, int year);

std::string dayOfWeek(int day);
// pre-condition: day has integer value 0, 1, 2, 3, 4, 5, or 6
// post-condition: the name of the day of week is returned as a std::. If day
has value 0, then SUNDAY is returned; 1, then MONDAY is returned; and so on.
```

For any function that the pre-condition and post-condition are missing, the students should add these conditions based their understanding of the function. The main function is given as following:

```
int main()
```

```
{
    using namespace std;
    int choice;
    int day, month, year;
    do
    {
        testMenu();
        cout << "Please choose from menu: ";
        cin >> choice;
        switch (choice)
        {
        case 1: // check if a given year is leap year
            cout << "Please enter a year: ";
            cin >> year;
            if (isLeapYear(year))
                cout << "Year " << year << " is a leap year" << endl;
            else
                cout << "Year " << year << " is NOT a leap year" << endl;
            break;

        case 2: // calculate the century value of a given year
            cout << "Please enter a year: ";
            cin >> year;
            cout << "The century value is: " << getCenturyValue(year) << endl;
            break;

        case 3: // calculate the year value of a given year
            cout << "Please enter a year: ";
            cin >> year;
            cout << "The year value is: " << getYearValue(year) << endl;
            break;

        case 4: // calculate the month value of a given month in a given year
            cout << "Please enter a year and month: ";
            cin >> year >> month;
            cout << "The month value is: " << getMonthValue(month, year) <<
endl;
            break;

        case 5: // calculate the day of week of a given date
            cout << "Please enter a year, a month, and a day : ";
            cin >> year >> month >> day;
            cout << "The day of the week is: " << dayOfWeek(month, day, year)
<< endl;
            break;
```

```
        case 6: // print out the name of a given day of week
            cout << "Please enter a day of week (0 for Sunday, 1 for Monday,
etc): ";
            cin >> day;
            cout << "The name of the day of the week is: " << dayOfWeek(day)
<< endl;
            break;

        case 7:
            cout << "Did you tested all functions yet ? if not, please rerun
the program\n";
            break;

        default:
            cout << "wrong option. Please choose from menu\n";
            break;
        }

        system("pause");

    } while (choice != 7);
}
```

Your job is to finish the implementation of all functions other than `main()` function.

Note: This project tells you a way to write functions and the code that manually test these functions. In the future, you may write code to automatically test the functions you or other people write.

Sample Run

Note: I use ⌨ symbol to show user inputs, you do **NOT** need to do ⌨ in your program.

```
***********************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
***********************************
Please choose from menu: ⌨1
Please enter a year: ⌨1900
```

```
Year 1900 is NOT a leap year
***********************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
***********************************
Please choose from menu: ⌨1
Please enter a year: ⌨2000
Year 2000 is a leap year
***********************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
***********************************
Please choose from menu: ⌨2
Please enter a year: ⌨2008
The century value is: 6
***********************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
***********************************
Please choose from menu: ⌨3
Please enter a year: ⌨2008
The year value is: 10
***********************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
```

```
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
************************************
Please choose from menu: ⌨4
Please enter a year and month: ⌨2016 10
The month value is: 0
************************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
************************************
Please choose from menu: ⌨5
Please enter a year, a month, and a day: ⌨2016 10 12
The day of the week is: 3
************************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
************************************
Please choose from menu: ⌨6
Please enter a day of week (0 for Sunday, 1 for Monday, etc): ⌨3
The name of the day of the week is: WEDNESDAY
************************************
* Test Menu *
* 1. isLeapYear *
* 2. getCenturyValue *
* 3. getYearValue *
* 4. getMonthValue *
* 5. dayOfWeek(month, day, year) *
* 6. dayOfWeek(day) *
* 7. Quit *
************************************
Please choose from menu: ⌨7
Did you tested all functions yet? if not, please rerun the program
Program ended with exit code: 0
```

Extra Credit

If you can validate user's input in your program, you can have 2 extra points. Input requirements are:

1. The input year is valid if it is later than year **1582**.
2. The input month is valid if it is between [1-12].
3. The input day is valid if it is a valid day base on the year and month.

If user input wrong date, please tell user an appropriate message.