

CIS 1111 Assignment:

2-Dimensional Arrays - Calendar

In this program, you will store a calendar month of dates (numbers starting at 1) in a two-dimensional array. Some months have 31 days, others 30, and February has 28 (29 if it's a leap year, which is a year divisible by 4).

Here is a list of Months, the number of Days in each month, and the Day on which each Month Starts, in 2016, 2017, 2018, and 2019:

<u>Months</u>	<u>Days</u>	<u>Day on which Month Starts</u>			
		<u>2016</u>	<u>2017</u>	<u>2018</u>	<u>2019</u>
January	31	Friday	Sunday	Monday	Tuesday
February	28*	Monday	Wednesday	Thursday	Friday
March	31	Tuesday	Wednesday	Thursday	Friday
April	30	Friday	Saturday	Sunday	Monday
May	31	Sunday	Monday	Tuesday	Wednesday
June	30	Wednesday	Thursday	Friday	Saturday
July	31	Friday	Saturday	Sunday	Monday
August	31	Monday	Tuesday	Wednesday	Thursday
September	30	Thursday	Friday	Saturday	Sunday
October	31	Saturday	Sunday	Monday	Tuesday
November	30	Tuesday	Wednesday	Thursday	Friday
December	31	Thursday	Friday	Saturday	Sunday

*except leap year

Given a month name, the name of the starting day of the week, and the year, write a program to store the date (number) of each day in the corresponding array location. Use a two-dimensional array called month of type int to store the generated month dates.

1. Declare month to be an array of type int of the proper size and dimensions (weeks and days), and initialize all elements to 0.
2. Handle user input as follows:
 - a. Ask the user for the month name, and match it to the appropriate number of days (hint: use parallel arrays)
 - b. If the month is February, you need to know the year, and test if it is a leap year to adjust February's number of days, if needed.
 - c. Ask the user which day of the week the month starts on, and match it to the proper day number (Sunday being day 1), to determine which position in the two-dimensional array to start storing dates (hint: use parallel arrays)
3. Use formatted output to display the month left-aligned, omitting "0" dates (see the output example below)
4. Use functions if desired. Allow your main program to loop, allowing the user to ask for an unlimited number of months.

```

C:\Users\Pat\Documents\Visual Studio CIS 1111 C++\Assn-Chapter 7-Calendar\Debug\Assn-Chap...
Enter the name of a month>February
Enter the year for February>2016
Enter the name of the day on which this month starts>Monday
    February    2016
     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

Enter the name of a month or enter stop to quit>August
Enter the year for August>2018
Enter the name of the day on which this month starts>Wednesday
    August    2018
           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 31

Enter the name of a month or enter stop to quit>February
Enter the year for February>2017
Enter the name of the day on which this month starts>Wednesday
    February    2017
           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

Enter the name of a month or enter stop to quit>stop

Press any key to continue . . .

```

Requirement	Point value
Used a 2-dimensional array of the proper type, size, and initialized to 0	2
Handled user input, accounted for invalid input, ask for the proper information as needed	3
Generated the month correctly, on the correct day of the week, with the correct number of days	6
Functions and variables were named clearly, and	4

code was well documented	
Output was clear and correctly formatted	5
Total	20