



CS 201R: Assignment II

Due: Sunday, Sept. 23

For this assignment we will use functions and arrays to create a calendar.

First, ask the user to enter a year (4-digit) and what week day does this year starts with (String). For example, the year 2018 starts with the week day “Friday”. In this case, the calendar should start from January 1 which is Friday, 2018.

Your program should produce a formatted calendar for the specified year and there should be a blank line after each month. Moreover, the month and the year should be centered over each month. Your program should reject any year that is less than 1000 or greater than 9999. Your program should handle a leap year correctly.

Development notes:

1. The tab character ('\t') can be used to align data into columns onscreen. We'll learn better ways of formatting output later during the semester.
2. You can use the predefined function “setw()” from the library “iomanip”. “setw()” allows you to pass a number and your output will be intended the number of spaces you passed to “setw”.
3. Before every month, print the week days: “Sun”, “Mon”, “Tue”, “Wed”, “Thu”, “Fri”, “Sat”.
4. Remember that the first day of each month has to be indented to the correct column. The last day of every month determines the first day of the next month. (Hint: Modulus operations make this calculation easy.)
5. To make it easier, you can convert the input string to upper case directly by using the predefined function “toupper()” or to lower case by using “tolower()”.
6. To check if a year is a leap year then:
 - I. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
 - II. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
 - III. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
 - IV. The year is a leap year (it has 366 days).
 - V. The year is not a leap year (it has 365 days).

7. You will need to define and implement the following functions:

- `void PrintOneMonth(int MonthNumber, int Year, int StartDayNumber);`
/*PRE: MonthNumber, Year, StartDayNumber all in range
POST: 1 month's calendar printed to output stream, with day headers
I/O: No input. Output to specified stream (default to screen)
USER INTERACTION: None.
ERROR HANDLING: None. Bad input will probably produce screwy output; that hasn't been tested. */
- `int DaysPerMonth(int MonthNumber, int Year);`
/*PRE: MonthNumber, Year in range
POST: returns days in that month & year, allowing for leap years.
I/O: None.
USER INTERACTION: None.
ERROR HANDLING: None. Bad input will probably produce bad results; that hasn't been tested. */
- `int GetStartDayNumber();`
/*PRE: None.
POST: Returns day of week as int (Sun = 0, Mon = 1, etc)
I/O: Screen & keyboard, no file I/O. User prompts, corrections on range
USER INTERACTION: Prompts user for day of week. Loops until valid correctly-spelled day is input.
Input is case-insensitive.
ERROR HANDLING: Loops until a valid day name is entered. Data is input as string so no problem if number is entered. */
- `int GetYear();`
/*PRE: None.
POST: Returns 4-digit number that is assumed to be a year, $1000 \leq Y \leq 9999$.
I/O: Screen/keyboard
USER INTERACTION: Prompts for year number. Expects integer input.
ERROR HANDLING: None. Entering non-numeric input will mess things up. */

8. You can add any parameter to the functions, also you can add more functions if you want.

Sample of output:

Enter year (must be in range 1000 <= Year <= 9999): 2018
What day of the week does Jan 1 fall on this year? monday

JANUARY 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	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			

FEBRUARY 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				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			

MARCH 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				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

And so on...

Submission:

Zip your project folder and submit it to Canvas by the deadline.