

Programming Assignment 2

Write a C program that prompts the user for a date (mm/dd/yyyy). The program should then take that date and use the formula on page 190 (see problem 2 in the textbook) to convert the date entered into a very large number representing a particular date. This very large number can be stored in a variable of type long integer, and is referred to in the **formula** below (and on page 190) as **N**. Your assignment is to take the value stored in variable **N** and, using the formula below, determine the number representing the day of the week that the date falls on. That is, if the number calculated by the algorithm is zero, then the day is Sunday. If the number calculated is a 1, then the day is Monday, etc.

The **formula** for calculating the numeric day of week is as follows:

$$\text{numeric day of week} = (\mathbf{N} - 621049) \text{ modulo } 7.$$

Your job is then to take that numeric day of the week (0 - 6), and output the day of the week in English text (such as "Monday"). See below for a sample program output.

Your program **requires** the following:

- function **calc_date_number**: which takes a date as an argument and returns a number representing the date entered.
- function **calc_day_number**: which takes as a parameter the number calculated in **calc_date_number**, and returns the number (0 - 6) representing the day of the week (Sunday - Saturday).
- function **display_day_of_week**: which takes as a parameter the **N** from the prior function. You must then take this numeric representation of the day of the week, and use it as an index into **a character string array**, to get the English representation of the day of the week. This can be done in function **main()**.

Hint:

Since we really haven't covered pointers to date, you can use the following variable declaration to solve this problem. Notice the "**character string array**" called **days**.

```
char *days[7] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
```

To use this in your program, simply use the **days** array to output your day of the week. For example, if

I wanted to output: **Today is Monday**, the printf statement would look like:

```
printf ("Today is %s\n", days[1]);
```

Assume the date entered by the user follows a particular valid format. That is, assume the user enters an **integer** (month) followed by a slash (/), another **integer** (day) followed by a slash (/), and another **integer** (year). (The input reference for this is in chapter 15.)

However, your program should ensure that the *month* entered is between 1 and 12, the *day* entered is between 1 and 31, and the *year* entered is greater than 1900, and that the actual

day exists in history taking into account leap years etc.

If the date is invalid, output error message and re-prompt, using the following

guidelines: If the *month* is invalid, output the following message:

Invalid month. Please re-enter date.

If the *day* is invalid, output the following message:

Invalid day. Please re-enter date.

If the *year* is invalid, output the following message:

Invalid year. Please re-enter date.

Other than the above requirements, you can be as creative as you'd like.

FYI -- This assignment is derived from Exercise #4 on page 191 in the textbook, where you will find the necessary formula for computing N.

You can also reference the section on formatted input and output in chapter 15.

The dialog with the user at a minimum should be as follows in this sample run:

Welcome to the Date to Day-of-Week program.

The program will give the day of the week for any date from 1/1/1900

Enter date (mm/dd/yyyy): **20/9/2015**

Invalid month. Please re-enter date.

Enter date (mm/dd/yyyy): **3/40/2015**

Invalid day. Please re-enter date.

Enter date (mm/dd/yyyy): **10/15/1800**

Invalid year. Please re-enter date. Year must be greater than 1900.

Enter date (mm/dd/yyyy): **10/15/2015**

10/15/2015 falls on a Thursday.

Thank you for using the Date to Day-of-Week program.

(Note: what the user types in is shown in **bold/blue** print for clarity only. Also, you do not need to output the 2 horizontal lines in the above segment.)