

## Kalendaryo

### 1 Description

**Kalendaryo** program that will generate the Gregorian calendar of a specific date. The format of the calendar may be: day, week, month, year.

#### 1.1 Date

The user will specify the date that he wants to view. The user will provide the date as **one integer** value, following the format *mmddyyyy*, e.g. 10082017<sup>1</sup>

#### 1.2 View

The calendar of the specified date will be shown. The format of the calendar shown depends on the current settings of View Format.

##### 1.2.1 Day View

In this view, the month (in words), day, year and weekday is displayed on screen. See Figure 1(a).

##### 1.2.2 Week View

In this view, the week that contains the specified date is shown. The display must contain month (in words) and year, days in that week and their specific weekdays. See Figure 1(b).

##### 1.2.3 Month View

In this view, the entire month that contains the specified date is displayed on the screen. See Figure 1(c).

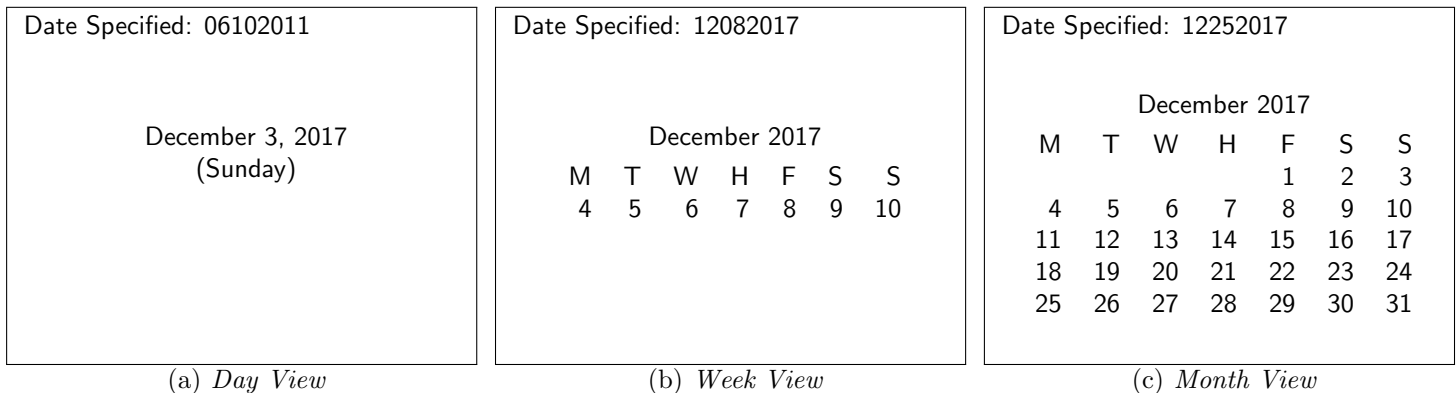


Figure 1: Different View Formats

##### 1.2.4 Year View with navigation

In this view, the calendar of the entire year that is specified in date is displayed on screen. If the specified date is 07191954, the calendar must show January 1, 1954 until December 31, 1954.

Since the 12-month calendar will not fit in one screen, navigation must be provided to allow the user to see the entire calendar. Assume that each screen (page) can only display 4 months (say January 1954 to April 1954),

---

<sup>1</sup>The month is represented using numbers 1 – 12, where 1 is for January, 2 for February, 3 for March, ...

navigation like previous and back allows the user to move from one page to another. When the user presses **Next**, the next 4 months of the calendar will be shown (i.e. May 1954 to August 1954). When the user presses **Previous**, the display should show the previous 4 months (January 1954 to April 1954).

### 1.3 First day of the Week

First day of the week may be specified by the user. There are only two options for this, either Sunday or Monday. The calendar displayed on the screen is affected by the specified first day of the week. Refer to Figure 2. By default, the first day of the week is set to Monday.

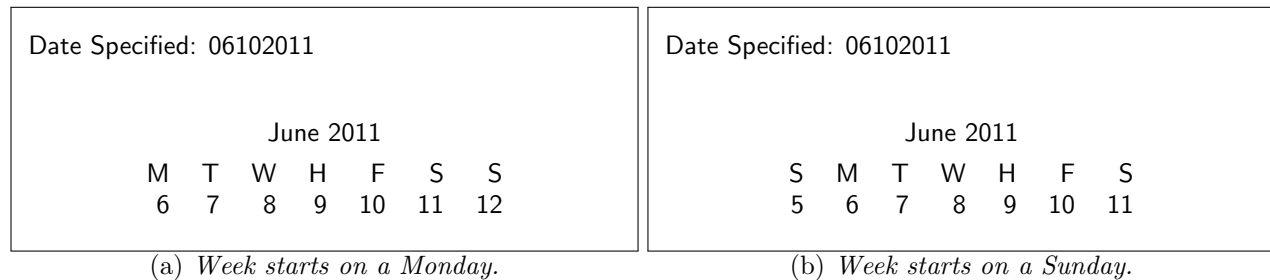


Figure 2: Calendar View Depends on the First Day of the Week

## 2 Requirements

**Kalendaryo is menu-driven.** The user may execute different tasks by selecting an option in the menu. After execution of each task, the main menu is presented again. The program will only end when the user chooses the Exit option in the main menu.

**Kalendaryo validates input data.** Data entered by the user is validated. Erroneous data should not cause the program to end or crash. When data provided is not valid, the user is informed, and the user will be asked to input until a valid entry is provided.

**Kalendaryo may be customized.** The user may change the settings of **Kalendaryo**. By default, the view format is **Day**, and the start of the week is **Monday**.

**Kalendaryo dates are aligned.** The calendar displayed on screen **must** be properly formatted. Specifically, each day must be right aligned to its corresponding weekday.

**Kalendaryo program has user-defined functions and uses of appropriate constructs** The implementation of this application must contain user-defined functions, wherever applicable and possible. Use the appropriate constructs while developing this application. No functions or use of brute force solutions will mean a *no submission of the project*.

## 3 Other Important Requirements

- The application must be developed using the C programming language. Be sure to successfully compile and run your programs in G302 and G306 labs before submitting.
- This is an individual project. Be sure to design and implement this project individually.
- For this application, you are **not allowed to use arrays**.

- Other functions and constructs not taught in class may be used in completing this application, provided that the student will learn them on their own and will be able to explain how and why they are used.

## 4 Submission

1. All projects must be submitted on or before **December 3, 2017 (Sunday)**.
2. All project submissions must be done through **Canvas ([dlsu.instructure.com](https://dlsu.instructure.com))**. Upload before the deadline **source code** and **function specifications**.
3. Unable to submit by the deadline will earn a grade of 0 for the machine problem. You may submit earlier than the deadline.
4. Unable to show up during the scheduled demo, or being unable to answer convincingly the questions during the demo, will merit a grade of 0 for the machine problem.
5. This is an **individual project**. Any form of cheating (working in collaboration, asking other people's help, copying any part of other's work, etc.) will merit a grade of 0.0 for the course and a discipline case.
6. Any requirement not fully implemented or instruction not followed will merit deductions.

## 5 Project Demo and Grading

1. All programs will be compiled first. Only successfully compiled projects will proceed with the demo.
2. No demo means *no submission of project*.
3. For grading scheme, see Table 1.
4. In instances wherein it is impossible to evaluate a project that has successfully compiled due to missing user input or calendar view, a grade of 0 may still be given.

Table 1: MP Grading Scheme

Screen design, user-friendly interface	10%
Menu-driven, continuous	10%
Program terminates normally	5%
Correct Weekday	10%
Calendar View - Day	5%
Calendar View - Week	10%
Calendar View - Month	15%
Error Checking	20%
Use of appropriate functions	15%
Total:	100%