**Lab 26**

**Problem 26.1**

This problem assumes TimeStruct\_t struct defined in Lab 25.

1. Create a TimeStruct\_t array **timeArray** of 2 elements, and initialize it to values of your choice using the array initializer syntax for structures.
2. Create 2 TimeStruct\_t variables **inTime** and **outTime**, and initialize them to 1st and 2nd elements of timeArray, respectively.
3. Display **inTime** and **outTime** using display\_time function.
4. Create a function apply\_time\_saving with the following prototype, which applies day light saving to all TimeStruct\_ts saved in **pTimeArray**. The day light saving is done by setting the clocks 1.5 hours back from the original time. The parameter ‘total’ contains total number of TimeStruct\_ts in **pTimeArray**.

void apply\_time\_saving(TimeStruct\_t **pTimeArray**[], int total);

**Problem 26.2**

Create a date structure named “DateStruct” containing year, month and day members. Now, create another structure named “DateTimeStruct” containing variables of both TimeStruct\_t and DateStruct as members. Now, input all values of members of both structs from user and display them in the following format:

The event occurred on 23-05-2017 at 03:49 pm

**Problem 26.3**

Using the DateTimeStruct in above problem, declare its array of three elements for saving 3 different alarms. Then, pass this array to a function named DisplayAlarmsInfo, which displays all three alarms information on separate lines in a similar format as Problem 25.2.