Write a complete C++ program, including a good comment in each function and in the main function, to do the following:

**Outline:** You are simulating a library, allowing the user to take out a book, place a hold on a book, return a book, and pay a fine. Since we did not cover arrays or strings, the program will be limited in its functionality. The main job of this program is to calculate and check due dates of a book. We may add functionality to the program in later assignments.

**Details:**

0. **Main:**

In the main function, you will print a library menu. Each choice will cause a different function to be called. The choices are:

   a. take out a book
   b. place a hold
   c. return a book
   d. pay fine
   e. exit

1. The function to take out a book asks the user for today's date and then adds two weeks to the date. It then prints to the screen the due date of the book, as mm/dd/yyyy. (Note: you cannot ask the user for the due date – you can only ask for today's date.) **To simplify the assignment, you may assume that every month of the year has 30 days, and every year has exactly 360 days.**

2. The function to place a hold is similar to the function to take out a book. The user is asked to enter today's date. The user is also asked how many weeks the hold should be relevant. The number of weeks entered is added to the current date, and the final date is printed to the screen.

3. The function to "return a book" to the library asks the user what today's date is, and what the return due date is. The two dates are compared, and if the book is overdue, a message is printed that there is a fine due.

4. The function to pay a fine asks the user two things: how much is owed, and how much user is paying. Remaining amount should be printed to the screen.
Design decisions:

A. The above four functions are mandatory; however, there are other functions that are necessary as well. Consider numbers 1 and 2 above. They both attempt to add a number of weeks to a date. A function should be written that adds **one week or one day** to a date. Then, each function can call this function in a loop the appropriate number of times.

The general principle is that any task that is performed more than once should be extracted and placed into its own function. Stay away from duplicating code.

B. You must use *reference parameters* in at least one function. A recommendation is to use reference parameters in the function that adds days to a date.

C. The key design decision of this program is how to store a date in your program. One option is to store a date as three integers: month, day, year. In this case, you will have to program if-else statements to add a number of days to a given date.

Another option, for e.g. Microsoft Excel uses this method, is to store the date as a single integer representing the number of days since a particular start date. This way it is easy to add days to a date, however, you will need two conversion functions – one to go from month/day/year to a number and vice versa.