```
// Logbook.cpp: implementation of the Logbook class.
//
//////////////////////////////////////////////////////////////////////

#include "Logbook.h"
#include <iostream>
#include <stdlib.h>
#include <stdio.h>

using namespace std;


//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////

Logbook::Logbook()
{
	setCurrentTime();
	initEntry();
}

Logbook::~Logbook()
{

}
```

```
Logbook::Logbook(int month, int year)
{
        setCurrentTime();

        // pre-lab
    logMonth = month;
    logYear = year;

        initEntry();
}

void Logbook::putEntry(int day, int value)
{
        // pre-lab
    entries[day] = value;
}

int Logbook::getEntry(int day) const
{
        // pre-lab
    return entries[day];
}

int Logbook::getMonth() const
{
```

```
        // pre-lab
    return logMonth;
}

int Logbook::getYear() const
{
        // pre-lab
    return logYear;
}

int Logbook::getDaysInMonth() const
{
        // pre-lab
    return DaysOfMonth[isLeapYear(logYear)][logMonth-1];
}

int Logbook::isLeapYear(int year) const
{
        if (((year % 4 == 0) && ((year % 100 != 0)) || (year % 400 == 0)))
                return 1;           // Leap year
        else
                return 0;           // Normal years
}

void Logbook::putEntry(int value)
```

```
{
	entries[(currentTime->tm_mday - 1)] = value;
}

void Logbook::setCurrentTime()
{
	// tm structure Used by asctime, gmtime, localtime, mktime,
	// and strftime to store and retrieve time information.

	// tm은 구조체로 시간 정보를 저장하고 불러오기 위해 asctime, gmtime, localtime, mktime,
	// 그리고 strftime 에 의해서 사용됩니다.

	time_t		now;
	time(&now);
	currentTime = localtime(&now);

	logYear = currentTime->tm_year + 1900;
	logMonth = currentTime->tm_mon + 1;
}

void Logbook::initEntry()
{
	int iDays = getDaysInMonth();
	for (int i = 0; i < iDays+1; i++)
		entries[i] = 0;
```

```
}


void Logbook::displayCalendar() const
{
	// In-lab

	printf("₩t₩t₩t₩t%d / %d₩n₩n", logMonth, logYear);
	cout << "Logbook:" << endl;
	cout << "SUN₩t";
	cout << "MON₩t";
	cout << "TUE₩t";
	cout << "WED₩t";
	cout << "THU₩t";
	cout << "FRI₩t";
	cout<<"SAT₩t"<< endl;

	int startDay = getDayOfWeek(1);

	for (int i = 0; i < startDay; i++)
		cout << "₩t";

	for (int day = 1; day <= getDaysInMonth(); day++)
	{
		cout << day << "    " << getEntry(day) << '₩t';
		if ((startDay+day) % 7 == 0)
```

```
                cout << endl;
        }
}

int Logbook::getDayOfWeek(int day) const
{
        tm          when;
        time_t      result;
        int nLeapYears = 0;
        int nDaysToMonth = 0;

        when = *currentTime;
        when.tm_mday = day;

        if ((result = mktime(&when)) != (time_t)-1)
        {
                for (int i = 1901; i < logYear; i++)
                {
                        if (isLeapYear(i))
                                nLeapYears++;
                }

                for (int i = 1; i < logMonth; i++)
                        nDaysToMonth += DaysOfMonth[isLeapYear(logYear)][i];

                return ((1 + when.tm_year + nLeapYears + nDaysToMonth + when.tm_mday) % 7);
```

```
        }
        else
        {
                return 0;
        }
}
```