

## Algorithms Level 4



26+ Years  
of Experience

# PROGRAMMING ADVICES

LEARN THE  
RIGHT WAY

**Mohammed Abu-Hadhoud**

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITIL®, MCPD, MCSD



حقوق النشر محفوظة، أسعار الكورسات في المنصة هي أسعار  
رمزية جدا، ارجو عدم نشر هذه الوثيقة لان نشرها سيمنعنا من  
الاستمرار في تقديم العلم للآخرين

ارجو عدم استخدام هذه الوثيقة من غير وجه حق لأنك ستحرم الاف  
الناس من التعلم

**[ProgrammingAdVICES.com](https://ProgrammingAdVICES.com)**



## Problem # 55/4 Solution Using C++

```
#include <iostream>
using namespace std;

struct stDate
{
    short Year;
    short Month;
    short Day;
};

bool isLeapYear(short Year)
{
    return (Year % 4 == 0 && Year % 100 != 0) || (Year % 400 == 0);
}

bool IsDate1BeforeDate2(stDate Date1, stDate Date2)
{
    return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ? (Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day < Date2.Day : false)) : false);
}

short NumberOfDaysInAMonth(short Month, short Year)
{
    if (Month < 1 || Month > 12)
        return 0;

    int days[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
    return (Month == 2) ? (isLeapYear(Year) ? 29 : 28) : days[Month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
    return (Date.Day == NumberOfDaysInAMonth(Date.Month, Date.Year));
}

bool IsLastMonthInYear(short Month)
{
    return (Month == 12);
}
```



## Problem # 55/4 Solution Using C++

```
stDate IncreaseDateByOneDay(stDate Date)
{
    if (IsLastDayInMonth(Date))
    {
        if (IsLastMonthInYear(Date.Month))
        {
            Date.Month = 1;
            Date.Day = 1;
            Date.Year++;
        }
        else
        {
            Date.Day = 1;
            Date.Month++;
        }
    }
    else
    {
        Date.Day++;
    }

    return Date;
}

short DayOfWeekOrder(short Day, short Month, short Year)
{
    short a, y, m;
    a = (14 - Month) / 12;
    y = Year - a;
    m = Month + (12 * a) - 2;
    // Gregorian:
    //0:sun, 1:Mon, 2:Tue...etc
    return (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m)
/ 12)) % 7;
}

short DayOfWeekOrder(stDate Date)
{
    return DayOfWeekOrder(Date.Day, Date.Month, Date.Year);
}
```



## Problem # 55/4 Solution Using C++

```
string DayShortName(short DayOfWeekOrder)
{
    string arrDayNames[] = {
        "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

    return arrDayNames[DayOfWeekOrder];
}

bool IsWeekEnd(stDate Date)
{
    //Weekends are Fri and Sat
    short DayIndex = DayOfWeekOrder(Date.Day, Date.Month,
Date.Year);
    return (DayIndex == 5 || DayIndex == 6);
}

bool IsBusinessDay(stDate Date)
{
    //Weekends are Sun, Mon, Tue, Wed and Thur

    /*
        short DayIndex = DayOfWeekOrder(Date.Day, Date.Month,
Date.Year);
        return (DayIndex >= 5 && DayIndex <= 4);
    */

    //shorter method is to invert the IsWeekEnd: this will save
    updating code.
    return !IsWeekEnd(Date);
}

short ReadDay()
{
    short Day;
    cout << "\nPlease enter a Day? ";
    cin >> Day;
    return Day;
}
```



```
short ReadMonth()
{
    short Month;
    cout << "Please enter a Month? ";
    cin >> Month;
    return Month;
}

short ReadYear()
{
    short Year;
    cout << "Please enter a Year? ";
    cin >> Year;
    return Year;
}

stDate ReadFullDate()
{
    stDate Date;

    Date.Day = ReadDay();
    Date.Month = ReadMonth();
    Date.Year = ReadYear();

    return Date;
}
```



## Problem # 55/4 Solution Using C++

```
stDate CalculateVacationReturnDate(stDate DateFrom, short
VacationDays)
{
    short WeekEndCounter = 0;

    for (short i = 1; i <= VacationDays; i++)
    {
        if (IsWeekEnd(DateFrom))
            WeekEndCounter++;

        DateFrom = IncreaseDateByOneDay(DateFrom);
    }
    //to add weekends
    for (short i = 1; i <= WeekEndCounter; i++)
        DateFrom = IncreaseDateByOneDay(DateFrom);

    //in case the return date is week end keep adding one day until
    you reach business day
    while (IsWeekEnd(DateFrom))
    {
        DateFrom = IncreaseDateByOneDay(DateFrom);
    }

    return DateFrom;
}

short ReadVacationDays()
{
    short Days;
    cout << "\nPlease enter vacation days? ";
    cin >> Days;
    return Days;
}
```



## Problem # 55/4 Solution Using C++

```
int main()
{
    cout << "\nVacation Starts: ";
    stDate DateFrom = ReadFullDate();

    short VacationDays = ReadVacationDays();

    stDate ReturnDate = CalculateVacationReturnDate(DateFrom,
    VacationDays);

    cout << "\n\nReturn Date: " <<
    DayShortName(DayOfWeekOrder(ReturnDate)) << " , "
    << ReturnDate.Day << "/" << ReturnDate.Month << "/" <<
    ReturnDate.Year << endl;

    system("pause>0");
    return 0;
}
```