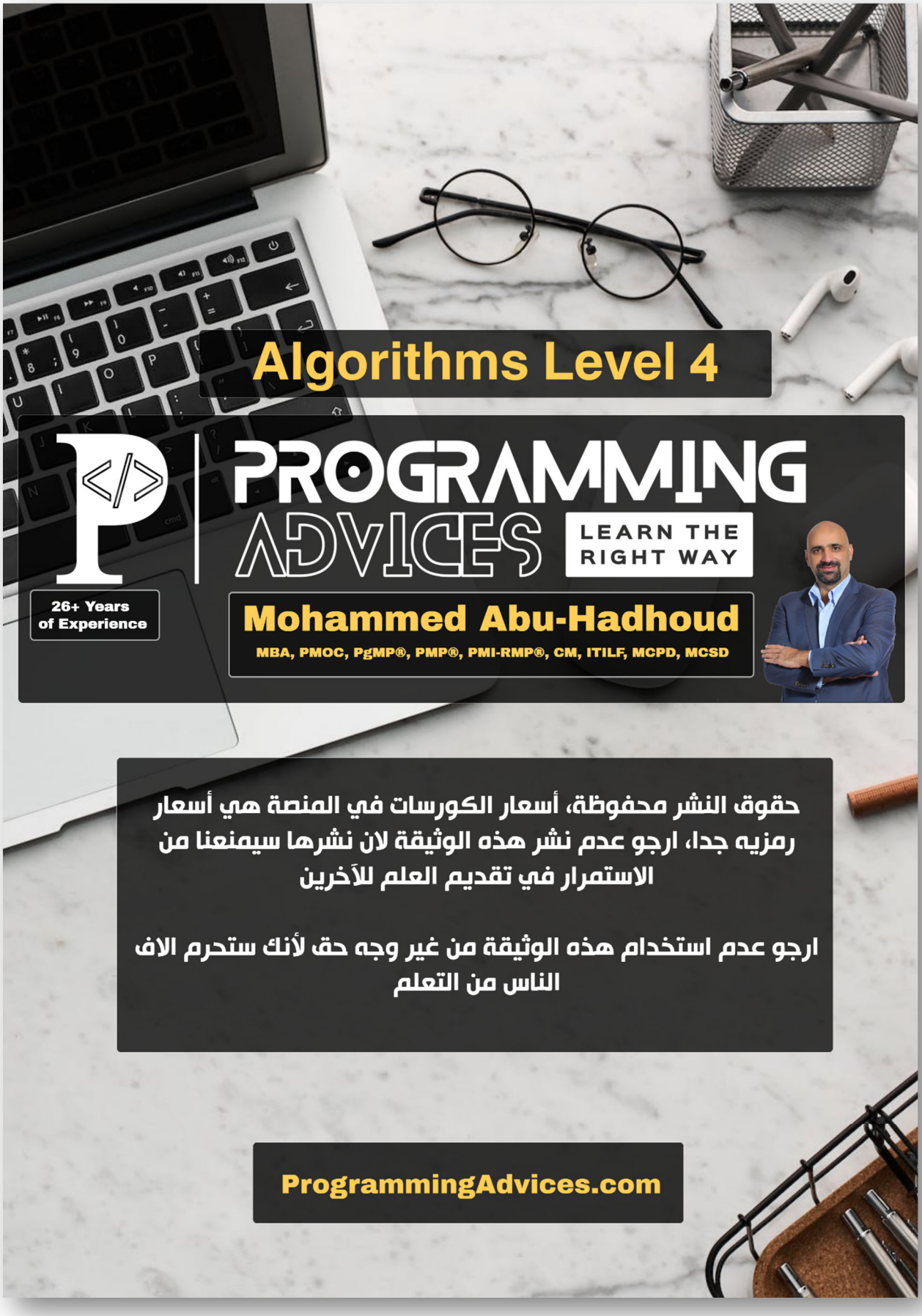
Algorithms Level 4
PROGRAMMING ADVICES
LEARN THE RIGHT WAY
26+ Years of Experience
Mohammed Abu-Hadhoud
MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD
حقوق النشر محفوظة، أسعار الكورسات في المنصة هي أسعار رمزيه جدا، ارجو عدم نشر هذه الوثيقة لان نشرها سيمنعنا من الاستمرار في تقديم العلم للآخرين
ارجو عدم استخدام هذه الوثيقة من غير وجه حق لأنك ستحرم الاف الناس من التعلم
ProgrammingAdvices.com

```cpp
#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);
}
```

```
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);

}
```

```
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;
}

short CalculateVacationDays(stDate DateFrom, stDate DateTo)
{

    short DaysCount = 0;
    while (IsDate1BeforeDate2(DateFrom, DateTo))
    {
        if (IsBusinessDay(DateFrom))
            DaysCount++;

        DateFrom = IncreaseDateByOneDay(DateFrom);
    }

    return DaysCount;
}
```

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

    cout << "\nVacation Ends: ";
    stDate DateTo = ReadFullDate();

    cout << "\nVaction From: " <<
DayShortName(DayOfWeekOrder(DateFrom)) << " , "
        << DateFrom.Day << "/" << DateFrom.Month << "/" <<
DateFrom.Year << endl;

    cout << "Vaction To: " << DayShortName(DayOfWeekOrder(DateTo))
<< " , "
        << DateTo.Day << "/" << DateTo.Month << "/" << DateTo.Year
<< endl;

    cout << "\n\nActucal Vacation Days is: " <<
CalculateVacationDays(DateFrom, DateTo);

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