

实用有趣的 C语言 程序

C Programming Language of Practical
and Interesting

杜树春 编著



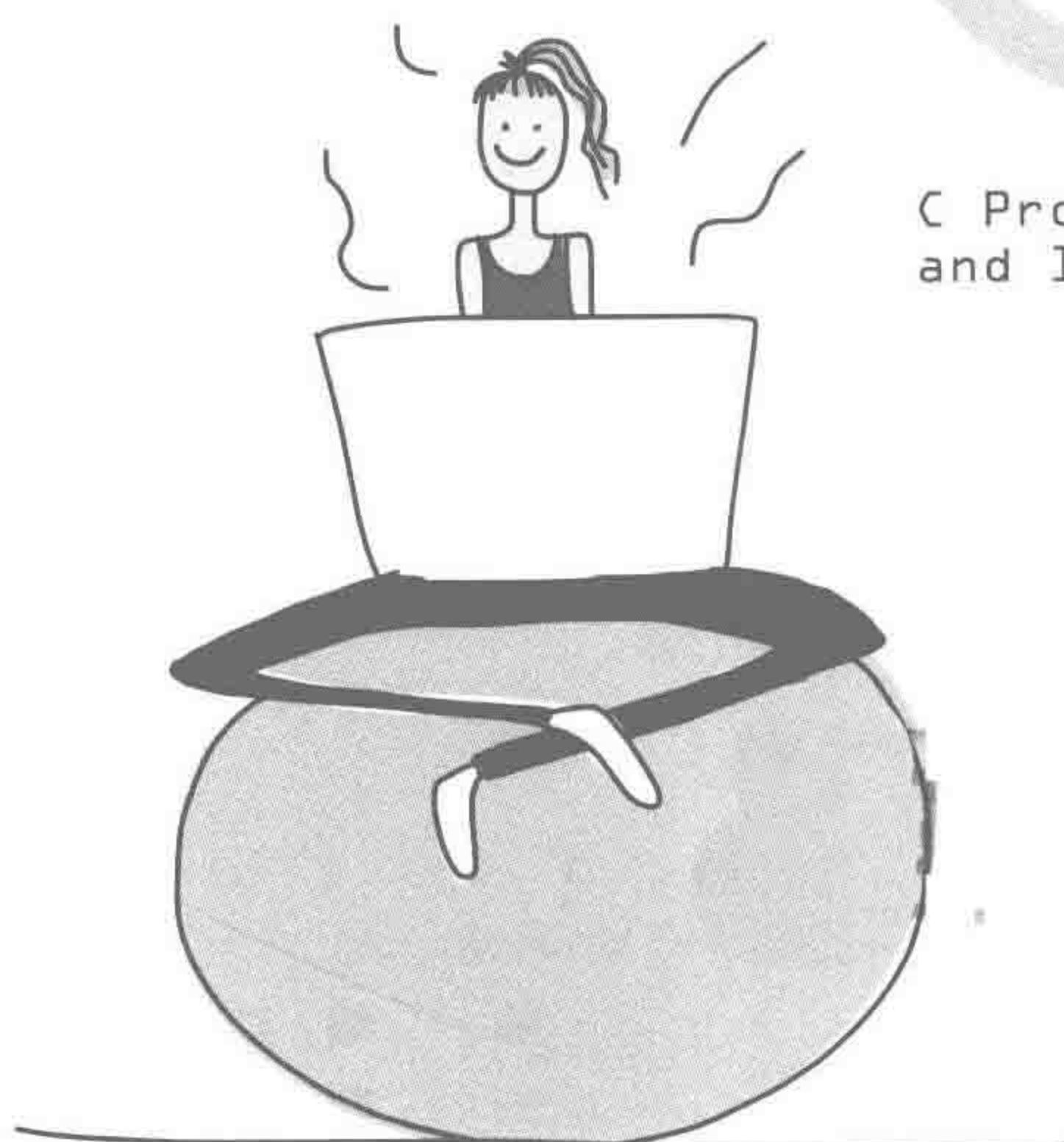
清华大学出版社



实用有趣的 C语言 程序

C Programming Language of Practical
and Interesting

杜树春 编著



清华大学出版社
北京

内 容 简 介

本书不是一本介绍如何使用 C 语言的书,而是一本实用、有趣的 C 语言程序集。

全书共分 9 章,第 1 章日历和日期,第 2 章大数计算,第 3 章极限和级数,第 4 章复数的运算,第 5 章矩阵计算,第 6 章解多元一次方程组,第 7 章解一元 N 次方程(上),第 8 章解一元 N 次方程(下),第 9 章其他程序。最后,有一个介绍 Visual C++ 6.0 上机操作的附录。

本书的特点是实践性强,书中每一个实例,读者无须重新输入程序,无须重新建立项目工程,只需用鼠标双击一下,程序就可以运行。

本书适合以下人员阅读或参考:一是学习 C 语言课程的大、中专及高等职业学校、中等职业学校的在校学生;二是使用 C 语言的广大工程技术人员;三是 C 语言编程的初学者;四是广大程序设计爱好者和 C 语言编程爱好者。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

实用有趣的 C 语言程序/杜树春编著. —北京:清华大学出版社,2017
ISBN 978-7-302-46219-4

I. ①实… II. ①杜… III. ①C 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 007866 号

责任编辑:文 怡 王 芳

封面设计:李召霞

责任校对:李建庄

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.25

字 数:371 千字

版 次:2017 年 2 月第 1 版

印 次:2017 年 2 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:073096-01



C 语言是国内外广泛使用的一种计算机语言。C 语言具有简洁、实用、代码质量高、可移植性强等优点,它同时具有高级语言和低级语言的特点,因此目前 C 语言已成为学习和使用人数最多的一种计算机语言。本书不是一本介绍如何使用 C 语言的书,而是一本实用、有趣的 C 语言程序集。

本书具有以下特点:①内容涉及面广,既包括初等数学方面的内容,也有高等数学方面的内容,还兼有其他的内容。②实例的选取兼顾了趣味性和实用性。③对于每一个 C 语言应用实例,本书都是先分析问题,提出大致思路,再给出解决此问题的完整的 C 语言程序清单,最后给出程序运行结果。

全书共分 9 章。第 1 章日历和日期,内容是根据日期求星期,求两个日期之间的天数,编写并打印出任一年的日历,以及农历和公历之间的相互转换等。第 2 章大数计算,这里的大数是指用普通计算器,包括计算机和手机上的计算器都无法正确计算出来的数。大数计算包括大数的加、减、乘、除、乘方、开平方、阶乘、排列和组合。第 3 章涉及极限、级数和圆周率等,用程序实现数列极限和函数极限的计算或验证;用程序实现级数前 N 项和的计算;以及用程序求出圆周率的精确值。第 4 章复数的运算,复数运算包括复数代数式与极坐标式的相互转换、复数求倒数、乘、除、乘方、开方、开 N 次方根、指数、对数、正弦、余弦等运算。第 5 章矩阵计算,包括求行列式的值、矩阵转置、矩阵相乘和矩阵求逆。第 6 章求解多元一次方程组,包括解二元、三元、四元、五元、六元一次方程组。第 7 章求解一元 N 次方程(上),包括用公式法解一元二次、一元三次、一元四次方程。第 8 章求解一元 N 次方程(下),包括用迭代法解一元五次、一元六次、一元七次和一元八次方程。第 9 章包括了没归类到前 8 章的其他若干程序。最后是介绍 Visual C++ 6.0 上机操作的附录。

本书提供的电子资料包的内容,仍是以书中章节为单位。在每一章(指第 1 章到第 9 章)下,都有一个章文件夹,每章下面有[例 $N.1$],[例 $N.2$], $\dots$,[例 $N.N$]的例文件夹,例文件夹内是这个例子的名称,打开名称文件夹,又有多个文件。其中,扩展名为 DSW 的文件是 Visual C++ 6.0 的工程文件;扩展名为 c 的文件是 Visual C++ 6.0 的源文件。在 Visual C++ 6.0 软件已安装在电脑中的前提下,双击具有 DSW 扩展名的文件就可进入 Visual C++ 6.0 软件集成环境,也就是 Visual C++ 6.0 的编辑、编译、连接、调试、运行环境。书中的所有例子都已在 Visual C++ 6.0 环境下调试通过,读者既可以原封不动地运行书中实

例,也可以改变输入参数运行,还可以修改程序(包括少量改动和大刀阔斧的改动)运行。

本书所用 C 语言编译集成环境是 Visual C++ 6.0,这个软件既可以用来编译、运行C++语言程序,也可以用来编译、运行 C 语言程序。本书所有例子都已在 Visual C++ 6.0 下通过调试。对于初次接触 C 语言的人,在读本书正文之前,可以先看一下介绍 Visual C++ 6.0 用法的附录。

本书的另一特点是实践性强。书中每一个实例,读者无须重新输入程序,无须重新建立项目工程,只需用鼠标双击一下,程序就可以运行。本书既适合初学者,也适合有一定编程基础的爱好者及专业技术人员。

本书适合以下人员阅读或参考:一是学习 C 语言课程的大、中专及高等职业学校、中等职业学校的在校学生;二是使用 C 语言的广大工程技术人员;三是 C 语言编程的初学者;四是广大程序设计爱好者和 C 语言编程爱好者。

由于编著者水平有限且时间仓促,书中难免存在不足,恳请读者批评指正。作者的电子邮箱为 dushuchun@263.net。

本书电子资料包请扫描二维码下载。



编著者

2016 年 12 月



第 1 章 日历和日期	1
1.1 给出年月日,计算该日是星期几	2
1.2 给出年份,计算该年元旦那天是星期几	3
1.3 计算某年某月某日是本年度第几天	4
1.4 求指定日期距离 1990 年 1 月 1 日的天数	6
1.5 给定出生年月日及现在年月日,计算天数	7
1.6 编制万年历	9
1.7 公历转农历	11
1.8 农历转公历	15
1.9 小结	19
第 2 章 大数计算	20
2.1 什么是大数计算	20
2.2 大数加法	21
2.3 大数减法	25
2.4 大数乘法	29
2.5 大数除法	31
2.6 大数阶乘 $n!$	33
2.7 大数乘方运算	36
2.8 大数开平方	38
2.9 大数排列	42
2.10 大数组合	44
2.11 小结	46
第 3 章 极限、级数和圆周率等	47
3.1 数列的极限和函数的极限	47

3.1.1	数列的极限和函数的极限的介绍	47
3.1.2	用 C 语言程序验证两种极限	49
3.2	级数、幂级数	52
3.2.1	级数的定义	52
3.2.2	用 C 语言程序验证级数的收敛和发散	53
3.3	圆周率 π	56
3.3.1	圆周率的来历	56
3.3.2	求圆周率的方法	57
3.4	自然对数的底 e	63
3.4.1	自然对数底的来历	63
3.4.2	用 C 语言程序计算 e 值	66
3.5	欧拉常数 γ	71
3.5.1	欧拉常数 γ 简介	71
3.5.2	用 C 语言程序计算 γ 值	72
3.6	小结	73
第 4 章	复数运算	74
4.1	复数简介	74
4.1.1	复数	74
4.1.2	复数的四种表示形式	75
4.1.3	复数的四则运算	76
4.1.4	复数的其他运算	78
4.2	复数运算程序 1	80
4.2.1	复数代数形式和极坐标形式间的相互转换程序	80
4.2.2	求代数形式复数的倒数程序	82
4.2.3	求任一复数平方根的程序	83
4.2.4	求两个代数形式复数之积的程序	84
4.2.5	求两个代数形式复数之商的程序	86
4.3	复数运算程序 2	87
4.3.1	求复数的乘幂程序	87
4.3.2	求复数的 N 次方根程序	88
4.3.3	求复数的指数程序	91
4.3.4	求复数的自然对数程序	92
4.3.5	求复数的常用对数程序	92
4.3.6	求复数的正弦程序	93
4.3.7	求复数的余弦程序	94
4.3.8	求复数的正切程序	95
4.4	小结	95

第 5 章 矩阵计算	96
5.1 矩阵简介	96
5.1.1 行列式	96
5.1.2 矩阵的加法、乘法和矩阵的转置	99
5.1.3 矩阵的除法——矩阵求逆	102
5.2 求行列式值的程序	103
5.2.1 求 3 阶行列式值的程序	103
5.2.2 求 4 阶行列式值的程序	105
5.2.3 求 5 阶行列式值的程序	106
5.2.4 求 6 阶行列式值的程序	107
5.3 矩阵转置程序	108
5.3.1 将 3 阶矩阵转置的程序	108
5.3.2 将 4 阶矩阵转置的程序	109
5.3.3 将 5 阶矩阵转置的程序	110
5.3.4 将 6 阶矩阵转置的程序	111
5.4 矩阵相乘程序	113
5.4.1 两个 3 阶实矩阵相乘程序	113
5.4.2 两个 4 阶实矩阵相乘程序	114
5.4.3 两个 5 阶实矩阵相乘程序	115
5.4.4 一个 4×5 矩阵与一个 5×3 矩阵相乘程序	116
5.5 矩阵求逆程序	117
5.5.1 求 2 阶矩阵的逆阵程序	117
5.5.2 求 3 阶矩阵的逆阵程序	120
5.5.3 求 4 阶矩阵的逆阵程序	122
5.5.4 求 5 阶矩阵的逆阵程序	124
5.5.5 求 6 阶矩阵的逆阵程序	126
5.6 小结	128
第 6 章 解多元一次线性方程组	129
6.1 多元一次方程组简介	129
6.2 解实系数多元一次方程组	131
6.2.1 求解实系数二元一次方程组	131
6.2.2 求解实系数三元一次方程组	133
6.2.3 求解实系数四元一次方程组	135
6.2.4 求解实系数五元一次方程组	137
6.2.5 求解实系数六元一次方程组	138
6.3 解复系数多元一次方程组	139
6.3.1 求解复系数二元一次方程组	139

6.3.2	求解复系数三元一次方程组	142
6.3.3	求解复系数四元一次方程组	143
6.3.4	求解复系数五元一次方程组	144
6.3.5	求解复系数六元一次方程组	145
6.4	小结	147
第7章	解一元 n 次方程(上)	148
7.1	实系数一元二次方程	149
7.2	用卡尔丹公式解实系数一元三次方程	151
7.3	用盛金公式解实系数一元三次方程	156
7.4	用谢国芳公式解实系数一元三次方程	161
7.5	实系数一元四次方程	167
7.6	复系数一元二次方程(1)	176
7.7	复系数一元二次方程(2)	181
7.8	小结	184
第8章	解一元 n 次方程(下)	185
8.1	实系数一元代数方程	185
8.1.1	求解实系数一元五次方程的根	186
8.1.2	求解实系数一元六次方程的根	190
8.1.3	求解实系数一元七次方程的根	191
8.1.4	求解实系数一元八次方程的根	192
8.2	复系数一元代数方程	194
8.2.1	求解复系数一元五次方程的根	194
8.2.2	求解复系数一元六次方程的根	198
8.2.3	求解复系数一元七次方程的根	200
8.2.4	求解复系数一元八次方程的根	201
8.3	小结	202
第9章	其他程序	203
9.1	乘法九九表 1	203
9.2	乘法九九表 2	204
9.3	杨辉三角形 1	205
9.4	杨辉三角形 2	206
9.5	斐波纳契数列	207
9.6	无符号整数的任意进制的转换	208
9.7	求指定区间 $[a, b]$ 内的勾股数组	210
9.8	百鸡问题	212
9.9	求 10000 以内的完全数	213

9.10	输入三角形三边求此三角形面积、外接圆半径和内切圆半径	214
9.11	围棋的棋局——千古无同局	215
9.12	求 1000 以内的素数	217
9.13	求 3~1000 以内的孪生素数	218
9.14	求 1000 以内的阿姆斯特朗数	219
9.15	计算气压随高度变化	220
9.16	小结	223
附录 Visual C++ 6.0 的上机操作		224
附.1	Visual C++ 6.0 的安装和启动	224
附.2	输入和编辑源程序	225
附.3	编译、连接和运行	228
附.4	编译和运行 C 语言程序的另一种方法	231
参考文献		233

第1章

日历和日期

公历是全世界通用的历法,以地球绕太阳的一周为一年,一年 365 天,分为 12 个月,1 月、3 月、5 月、7 月、8 月、10 月和 12 月为 31 天,2 月为 28 天,其余月份为 30 天。事实上地球绕太阳一周共 365 天 5 小时 48 分 46 秒,比公历一年多出 5 小时 48 分 46 秒。为使年误差不累积,公历年用闰年法来消除年误差,由于每年多出 5 小时 48 分 46 秒,每 4 年累计多出 23 小时 15 分 4 秒,接近 1 天。天文学家就规定每 4 年有一个闰年,把 2 月由 28 天改为 29 天,凡是公历年能被 4 整除的那一年就是闰年,但是这样一来每 4 年又少了 44 分 56 秒。为了更准确地计时,天文学家又规定凡能被 100 整除的年份,只有能被 400 整除才是闰年,即每 400 年要减掉 3 个闰年(例如,1700 年、1800 年和 1900 年不是闰年,2000 年才是闰年),经过这样处理后,实际上每 400 年的误差只有 2 小时 53 分 20 秒,已相当准确了。

根据以上情况,某一年是否为闰年应按以下规则计算:闰年应能被 4 整除(如 2016 年是闰年,2015 年不是闰年),但不是所有能被 4 整除的年份都是闰年。在能被 100 整除的年份中,只有同时能被 400 整除的年份才是闰年(如 2000 年是闰年),能被 100 整除而不能被 400 整除的年份(如 1800 年、1900 年,2100 年)不是闰年。

农历与公历不同,农历把月亮绕地球一周作为一月,一年 12 个月,总共 354 天或 355 天。而天气的冷热变化一次是 365 天,它和回归年(公历年)一致。这样农历和公历一年要相差 11 天或 10 天,3 年就是 1 个月多。为使农历能适应天气冷热变化的周期,就在第 3 年加上 1 个月,这一年就有 13 个月,加上的这个月叫“闰月”,1 年就是 384 或 385 天了。“闰月”也分为大月和小月,闰大月 30 天,闰小月 29 天。可以在任何一个月闰,既可闰大月,也可闰小月。农历采用 19 年加 7 个闰月的办法,即“十九年七闰法”,把回归年与农历年很好地协调起来。农历是中国传统文化的代表之一,并与农业生产联系密切,中国人民特别是广大农民十分熟悉并喜爱农历。

公历与农历是我国目前并存的两种历法,各有其固有的规律,农历与月球的运行相对应,其影响因素多。它的大小月和闰月与天体运行有关计算十分复杂,且每年都不一致。因此要用计算机或单片机实现公历与农历的转换,用查表法是最方便实用的办法。

1.1 给出年月日,计算该日是星期几

1. 编程说明

根据历法原理,按照下面的公式计算,就可以知道某年某月某日是星期几了。这个公式是:

$$S = x - 1 + \left[\frac{x-1}{4} \right] - \left[\frac{x-1}{100} \right] + \left[\frac{x-1}{400} \right] + C$$

这里, x 是公元的年数, C 是从这一年的元旦算到这天为止(连这一天也在内)的日数。式中, $\left[\frac{x-1}{4} \right]$ 表示为 $\frac{x-1}{4}$ 的整数部分,也就是说,三个分数式只取商数的整数部分,余数略去不计。求出 S 后,再用 7 除。如果恰能除尽,这一天一定是星期日;若余数是 1,这一天是星期一;余数是 2,这一天就是星期二。以此类推。程序中,占大部分篇幅的程序是在计算 C , 算 C 之前,还有判断所求年份是否是闰年,这样决定 2 月份是 28 天还是 29 天。

例如,要求算 2016 年的 1 月 2 日是星期几。

$$\begin{aligned} S &= 2016 - 1 + \left[\frac{2016-1}{4} \right] - \left[\frac{2016-1}{100} \right] + \left[\frac{2016-1}{400} \right] + 2 \\ &= 2505 \end{aligned}$$

而 $2505 \% 7 = 357$ 余 6,这说明 2016 年的 1 月 2 日是星期六。

2. 程序清单

```

/*****
/* 1.c 给出年月日,计算该日是星期几 */
/*****
#include <stdio.h>
main()
{
    int a,b,c,d,s,z;
    printf("请输入年月日 yyyy,mm,dd: ");
    scanf("%d,%d,%d",&a,&b,&c);
    if((a%4==0&&a%100!=0)|| (a%400==0))
    {
        if(b==1)d=c;
        if(b==2)d=c+31;
        if(b==3)d=c+60;
        if(b==4)d=c+91;
        if(b==5)d=c+121;
        if(b==6)d=c+152;
        if(b==7)d=c+182;
        if(b==8)d=c+213;
        if(b==9)d=c+244;
    }
}

```



```

    if(b == 10)d = c + 274;
    if(b == 11)d = c + 305;
    if(b == 12)d = c + 335;
    }
    else{ if(b == 1)d = c;
    if(b == 2)d = c + 31;
    if(b == 3)d = c + 59;
    if(b == 4)d = c + 90;
    if(b == 5)d = c + 120;
    if(b == 6)d = c + 151;
    if(b == 7)d = c + 181;
    if(b == 8)d = c + 212;
    if(b == 9)d = c + 243;
    if(b == 10)d = c + 273;
    if(b == 11)d = c + 304;
    if(b == 12)d = c + 334; }
    s = a - 1 + (a - 1)/4 - (a - 1)/100 + (a - 1)/400 + d;
    z = s % 7;
    if(z == 0)printf(" 这一天为星期天。 \n");
    if(z == 1)printf(" 这一天为星期一。 \n");
    if(z == 2)printf(" 这一天为星期二。 \n");
    if(z == 3)printf(" 这一天为星期三。 \n");
    if(z == 4)printf(" 这一天为星期四。 \n");
    if(z == 5)printf(" 这一天为星期五。 \n");
    if(z == 6)printf(" 这一天为星期六。 \n");
}

```

3. 程序运行

【例 1.1】 给出年月日,计算该日是星期几。

解: 打开“给出年月日,计算该日是星期几?”文件夹,双击扩展名为 DSW 的文件,进入 Visual C++ 6.0 编译环境,编译程序并执行,根据屏幕提示,输入年月日,运行结果如下:

```

请输入年月日 yyyy,mm,dd:2016,05,18
这一天为星期三。

```

由运行结果可见,当你输入 2016 年 5 月 18 日时,程序告诉你这一天是星期三。

1.2 给出年份,计算该年元旦那天是星期几

1. 编程说明

根据历法原理,设 a 年元旦是星期 s (取值为 0~6,其中 0 为星期日),整数 s 的计算公式是:

$$s = \left(a + \left[\frac{a-1}{4} \right] - \left[\frac{a-1}{100} \right] + \left[\frac{a-1}{400} \right] \right) \% 7$$

这里, a 是公元的年数, $\left[\frac{a-1}{4} \right]$ 表示为 $\frac{a-1}{4}$ 的整数部分,也就是说,三个分数式只取商数的整数部分,余数略去不计。求出小括号内的数值后,再用 7 除,所得结果就是 s 。如果恰能除尽,这一天一定是星期日;若余数是 1,这一天是星期一;余数是 2,这一天就是星期二;

以此类推。例如,要求算 2015 年的元旦(1 月 1 日)是星期几。

$$\begin{aligned}
 s &= \left(a + \left[\frac{a-1}{4} \right] - \left[\frac{a-1}{100} \right] + \left[\frac{a-1}{400} \right] \right) \% 7 \\
 &= \left(2015 + \left[\frac{2015-1}{4} \right] - \left[\frac{2015-1}{100} \right] + \left[\frac{2015-1}{400} \right] \right) \% 7 \\
 &= 2503 \% 7
 \end{aligned}$$

而 $2503 \% 7 = 357$ 余 4,这说明 2015 年的 1 月 1 日是星期四。

2. 程序清单

```

/ ***** /
/* 1.c  给出年份,计算该年元旦是星期几?          */
/ ***** /
#include <stdio.h>
main()
{
    int a,s,z;
    printf("请输入年份 yyyy: ");
    scanf("%d",&a);
    s = a + (a-1)/4 - (a-1)/100 + (a-1)/400;
    z = s % 7;
    if(z == 0)printf("元旦这一天为星期天。\\n");
    if(z == 1)printf("元旦这一天为星期一。\\n");
    if(z == 2)printf("元旦这一天为星期二。\\n");
    if(z == 3)printf("元旦这一天为星期三。\\n");
    if(z == 4)printf("元旦这一天为星期四。\\n");
    if(z == 5)printf("元旦这一天为星期五。\\n");
    if(z == 6)printf("元旦这一天为星期六。\\n");
}

```

3. 程序运行

【例 1.2】 给出年份,计算该年元旦那天是星期几。

解: 打开“给出年份,计算该年元旦那天是星期几?”文件夹,双击扩展名为 DSW 的文件,进入 Visual C++ 6.00 编译环境,编译程序并执行,根据屏幕提示,输入年份,运行结果如下:

```

请输入年份yyyy:2016
元旦这一天为星期五。

```

由运行结果可见,当你输入 2016 年时,程序告诉你 2016 年的元旦这一天是星期五。

1.3 计算某年某月某日是本年度第几天

1. 编程说明

在公历的一年中,除 2 月份以外,每月的天数都是固定的。已知某年某月某日,计算该日子是本年度的第几天,首先判断该年份是否为闰年,求出 2 月份是 28 天还是 29 天;再把

已过月份每个月的日子加起来,最后加当月的日期数。sum_day 计算输入日期的天数,leap 函数返回是否是闰年的信息。

2. 程序清单

```

/*****
/* 1.c 给出年月日,计算该日是该年第几天? */
/*****
#include <stdio.h>
void main()
{
    int sum_day(int, int);
    int leap(int year);
    int year, month, day, days;
    printf("\n");
    printf("请输入所求日期(年(4位),月(2位),日(2位)): ");
    scanf("%d, %d, %d", &year, &month, &day);
    printf("%d, %d, %d", year, month, day);
    days = sum_day(month, day);
    if (leap(year) && month >= 3)
        days = days + 1;
    printf("是本年第 %d 天。 \n", days);
}
int sum_day(int month, int day)
{
    int day_tab[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int i;
    for (i = 1; i <= month - 1; i++)
        day += day_tab[i];
    return(day);
}
int leap(int year)
{
    int leap;
    leap = year % 4 == 0 && year % 100 != 0 || year % 400 == 0;
    return(leap);
}

```

3. 程序运行

【例 1.3】 给出年月日,计算该日是该年第几天。

解: 打开“给出年月日,计算该日是该年第几天”文件夹,双击扩展名为 DSW 的文件,进入 Visual C++ 6.0 编译环境,编译程序并执行之,根据屏幕提示,输入年月日,运行结果如下:

```

请输入所求日期<年<4位>,月<2位>,日<2位>>:2016,5,18
2016.5.18是本年第139天。

```

由运行结果可见,2016 年 5 月 18 日是 2016 年的第 139 天。

1.4 求指定日期距离 1990 年 1 月 1 日的天数

1. 编程说明

已知某年某月某日,计算该日子距离 1990 年 1 月 1 日的天数,由两部分组成,一是已知年到 1990 年是几年并算出这些年的天数之和,二是算出已知月已知日距当年 1 月 1 日的天数,这两部分天数之和即为所求。

2. 程序清单

```

/ ***** /
/* 1.c 指定年月日求距离 1990 年 1 月 1 日的天数及小时数 */
/ ***** /
#include <stdio.h>
typedef struct date {
    int year;
    int month;
    int day;
}DATE;
int countday(DATE);
int runyear(int);
void main()
{
    DATE today;
    int totalday;
    printf("请输入指定年月日(yyyy,mm,dd): ");
    scanf("%d, %d, %d",&today.year,&today.month,&today.day);
    totalday = countday(today) - 1;
    printf("总天数是 %d 天。\\n",totalday);
    printf("总小时数是 %d 。\\n",24 * totalday);
}
int countday(DATE currentday)
{
    int permonth[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
    int totalday = 0, year, i;
    for (year = 1990; year < currentday.year; year++)
    {
        if (runyear(year))
            totalday = totalday + 366;
        else
            totalday = totalday + 365;
    }
    if (runyear(currentday.year))
        permonth[2] += 1;
    for (i = 0; i < currentday.month; i++)
        totalday += permonth[i];
    totalday += currentday.day;
    return(totalday);
}

```



```

int runyear(int year)
{
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
        return 1;
    else
        return 0;
}

```

3. 程序运行

【例 1.4】 指定年月日求距离 1990 年 1 月 1 日的天数及小时数。

解：打开“指定年月日求距离 1990 年 1 月 1 日的天数及小时数”文件夹，双击扩展名为 DSW 的文件，进入 Visual C++ 6.0 编译环境，编译程序并执行，根据屏幕提示，输入年月日，运行结果如下：

```

请输入指定年月日(yyyy,mm,dd):2016,5,19
总天数是 9635 天。
总小时数是 231240 。

```

由运行结果可见，1990 年 1 月 1 日到 2016 年 5 月 19 日共有 9635 天或 231240 小时。

1.5 给定出生年月日及现在年月日，计算天数

1. 编程说明

本节的问题，与上一节的问题类似。上一节是求指定年月日距离 1990 年 1 月 1 日的天数；本节是给定出生年月日及现在年月日，计算天数。本节计算也是由两部分组成，一是现在年到出生年是几年并算出这些年天数之和，二是算出现在月日距出生月日的天数——如果现在月日比出生月日大，所得天数是正的；反之，如果现在月日比出生月日小，所得天数是负的。这两部分天数之和即为所求。

2. 程序清单

```

/*****
/* 1.c 给定出生年月日及现在年月日，计算天数 */
*****/
#include <stdio.h>
void main()
{
    int sum_day(int, int);
    int leap(int year);
    int year1, month1, day1, days1;
    int year, month, day, days, j, dayss = 0;
    printf("请输入出生年月日(yyyy,mm,dd): ");
    scanf("%d, %d, %d", &year1, &month1, &day1);
    printf("请输入现在年月日(yyyy,mm,dd): ");
    scanf("%d, %d, %d", &year, &month, &day);
    days1 = sum_day(month1, day1);
    if (leap(year1) && month1 >= 3)
        days1 = days1 + 1;
    days = sum_day(month, day);
}

```