

电子信息与通信学院

实 验 报 告

|  |  |
| --- | --- |
| 实验名称 | 课程综合练习 |
| 课程名称 | 计算机基础  与程序设计(C) |

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 牛思途 | 学号 | U202411211 |

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 2024.12 | 地点 | 华中科技大学 |

|  |  |  |  |
| --- | --- | --- | --- |
| 成绩 |  | 教师 | 刘威 |

# 实验目的

完成xxx系列代码（日历系列、大数计算系列，选择其中之一）。

# 实验环境

操作系统：Windows 10

编程工具：CodeBlocks 16.01

# 实验一

## 实验任务

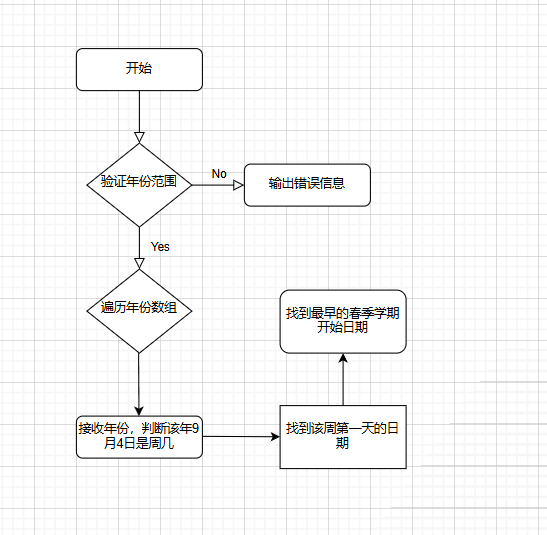
用数组改写打印华中科技大学校历的程序，具备打印校历的能力，比较多个年份的校历秋季校历的首日，寻找开学最早的那一年

约定秋季学期从当年9月4日所在周的周一开始，到第二年1月20日之前一周结束；约定春季学期从次年2月15日之后一周开始，7月第一周结束。

用两个只读的全局的一维数组记录不同月份的天数长度基于数组改进现有的日历程序，改进计算目期的函数

## 实验步骤

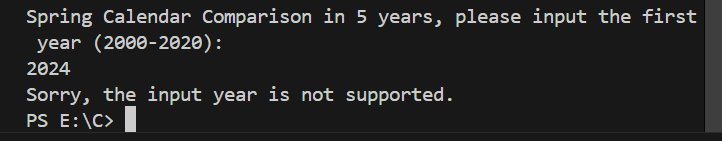
* 再完成这个实验时，我通过试验任务设计了多个函数，因为是不同年份的判断，所以我设计isleapyear函数用以判断是否是闰年，用getdayseqonjan1获取每年第一天是周几，getDaySeq获取指定日期是该年的第几天。getWeekSeqOfYear获取指定日期是该年的第几周。getMonth、getDay从年的第几天获取月份和日期。getDaySeqOfNeek获取指定年份的第几天是星期几。getThisMonday、getThisSunday：获取指定年份的某个日期是星期一还是星期日。printOneDay、printOneWeek：打印一天或一周的日历。setYearArray：设置年份数组。
* 而main函数用以调用这些函数，从而完成目标。

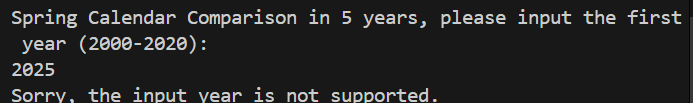


## 代码测试

### 测试点的测试结果

测试越界值，试值输入后结果合理：





对多个函数的复合测试，试值输入后结果合理。

故测试通过

## 实验结论

代码可以达到功能目标。

## 实验总结

在开始时我计算日期的逻辑有误，导致数组访问越界，产生了一个非常滑稽的结果，在修改getdayseqofweek函数后，返回值能够正确索引Month\_NORMAL\_YEAR 或 Month\_LEAP\_YEAR。

在写printoneweek函数时，开始总是print不出来想要的格式，后来重新改了一遍输出逻辑，终于能对齐。

我在开始时没有验证输入年份的范围，在输入一个较小的年份时导致了程序异常。

# 实验二

## 实验任务

用三维数组记录多年的日期数据，查找某人的生日，并打印该生日所在周的周历，计算其在周末过生日的次数

## 实验步骤

为了完成实验任务，以处理不同年份的日期数据。isLeapYear 函数，用以判断指定年份是否为闰年。getDaySeqOnJan1 函数，获取每年第一天是星期几。getDaySeq 函数，获取指定日期是该年的第几天。实现 getWeekSeqOfYear 函数，获取指定日期是该年的第几周。getMonth 和 getDay 函数，从年的第几天获取月份和日期。getDaySeqOfWeek 函数，获取指定年份的第几天是星期几。getThisMonday 和 getThisSunday 函数，获取指定年份的某个日期是星期一还是星期日。printOneDay 和 printOneWeek 函数，打印一天或一周的日历。setYearArray 函数，设置年份数组。initialDays 函数初始化三维数组，记录多年的日期数据。

在主函数 main 中，提示用户输入生日的月份和日期。对于找到的每个生日日期，使用 printOneWeek 函数打印该生日所在周的周历。检查生日日期是否在周末（星期六或星期日）。统计并记录周末过生日的次数，最后输出结果：打印总的生日次数以及在周末过生日的次数。

4.3实验结论：代码可以达到功能目标

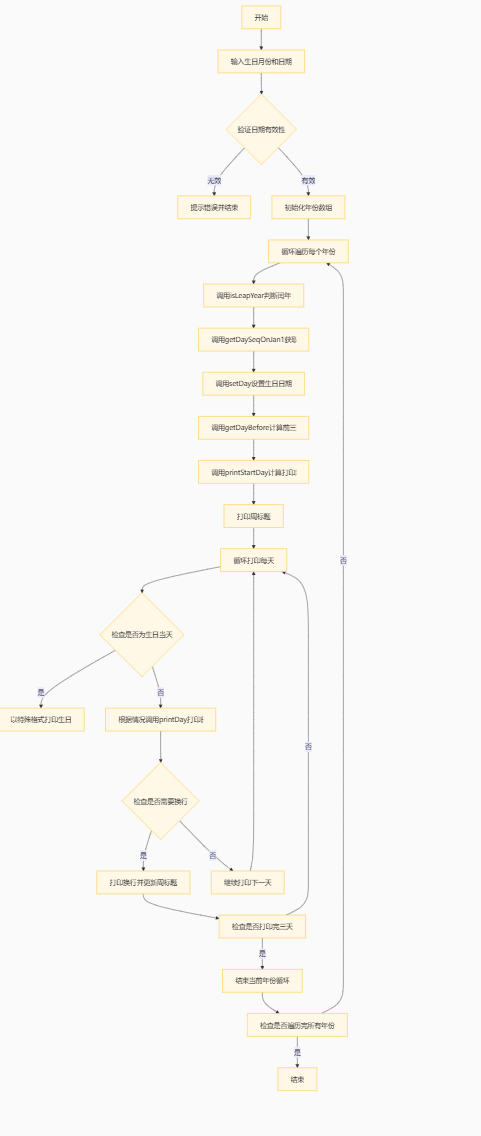
# 实验三

## 实验任务

假定某生日趴需要三天时间准备，输入某人的生日，通过日期偏移计算获得前三天并打印相关周历**。**用日期结构体记录单一日期的所有数据属性**。**

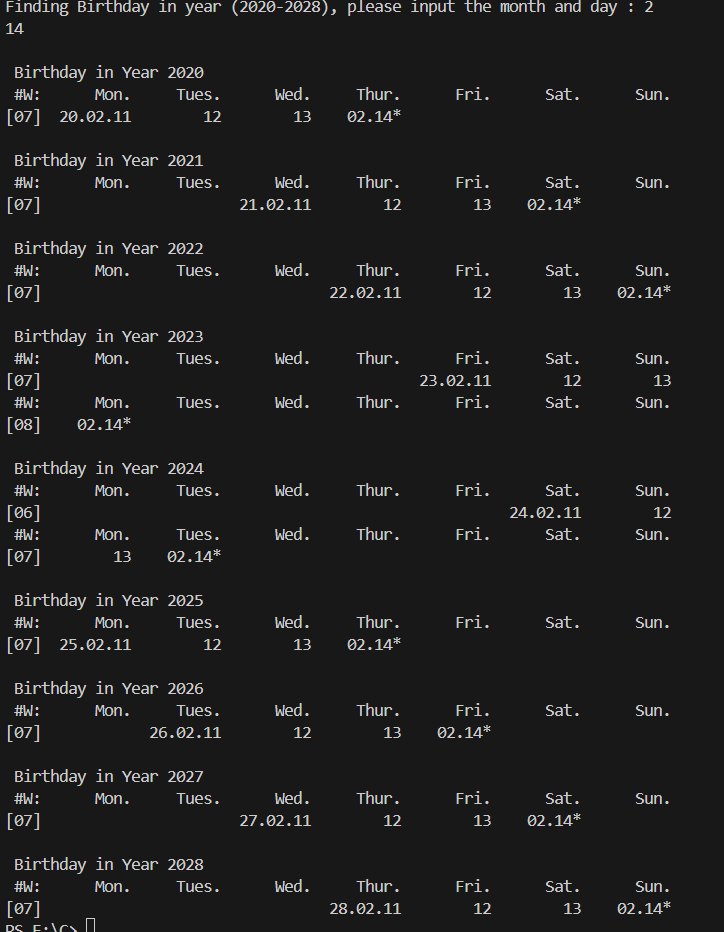
## 5.2 实验步骤

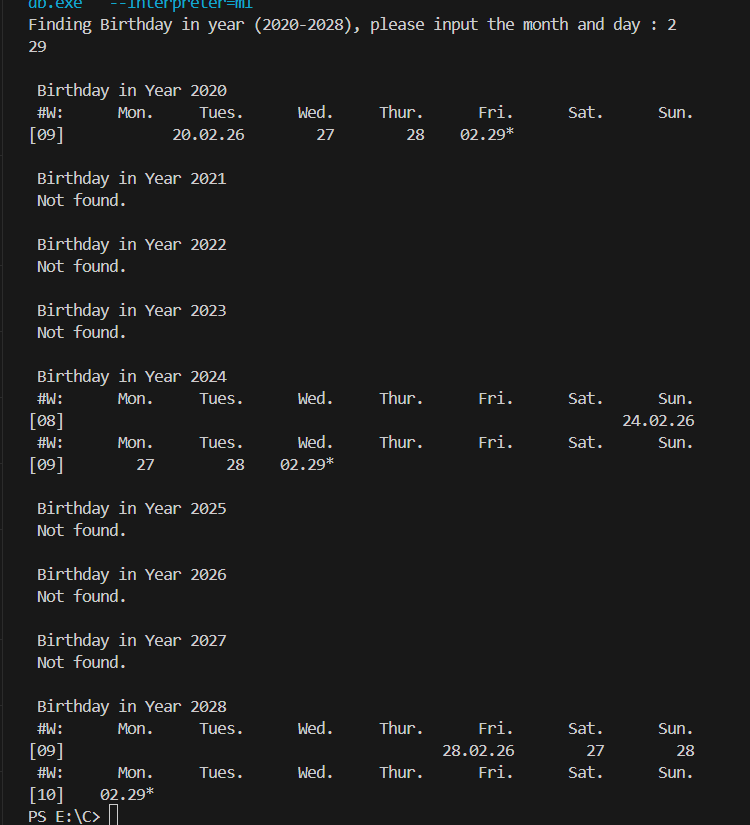
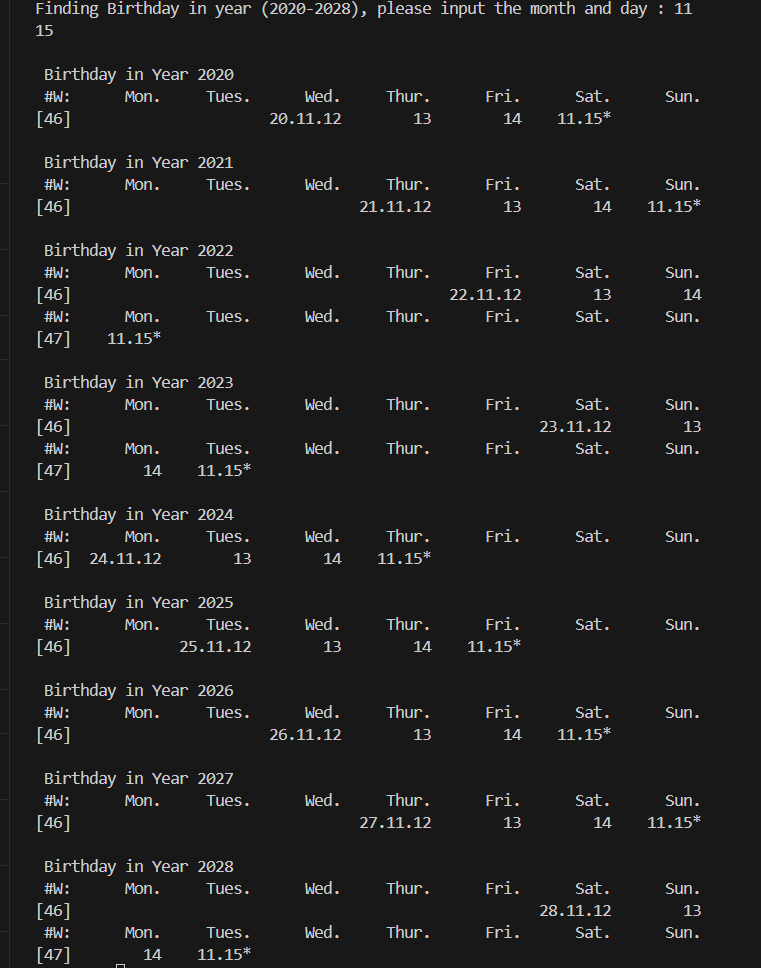
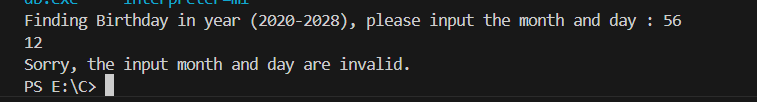
设计闰年判断函数：我创建了一个名为isLeapYear的函数，用于判断给定的年份是否为闰年。这个函数通过检查年份是否能被4整除但不能被100整除，或者能被400整除来确定是否为闰年。获取一年第一天是周几：我设计了一个名为getDaySeqOnJan1的函数，用于计算并返回给定年份的第一天是星期几。这个函数通过累加前一年的天数并考虑闰年的影响来计算。获取指定日期是该年的第几天：我实现了一个名为getDaySeq的函数，它接受年份、月份和日期作为参数，并返回该日期是该年的第几天。获取指定日期是该年的第几周：我创建了一个名为getWeekSeqOfYear的函数，它计算并返回给定日期是该年的第几周。从年的第几天获取月份和日期：我设计了两个函数getMonth和getDay，它们接受年份和一年中的第几天作为参数，并返回相应的月份和日期。获取指定年份的第几天是星期几：我实现了一个名为getDaySeqOfWeek的函数，它计算并返回给定年份的第几天是星期几。获取指定年份的某个日期是星期一还是星期日：我创建了两个函数getThisMonday和getThisSunday，它们根据给定的年份和日期，返回该日期是星期一还是星期日。打印一天或一周的日历：我设计了两个函数printOneDay和printOneWeek，它们分别用于打印指定日期的一天日历和一周日历。设置年份数组：我实现了一个名setYearArray的函数，它根据给定的年份数量、最小年份和最大年份，生成一个包含这些年份的数组。主函数调用：在main函数中，我调用了上述所有函数，以实现用户输入生日日期后，程序能够计算并打印出生日前三天的日历信息。



5.3代码测试

经全局多个样本测试，所有函数功能正常。





## 5.4实验结论

代码可以达到功能目标。

5.5实验总结

该程序设计以结构体的形式储存了包括年月日周序数在内的多个参数，体现了结构体能够存储多种数据的特性。

# 本课程学习总结

学习C语言的历程和总结

这个学期我刚开始涉及编程语言学，与c语言的相遇从printf(“helloWorld!”)开始，至今仍未结束，以后大概缘分会越来越紧密吧。眼睁睁看着自己从一开始的懵懂到能完成简单的三十行代码，再到现在能够运用多函数解决复杂问题。有进步的喜悦也有感慨。虽然自己的“码力”仍旧一般，不过也算是入了编程的门槛。下学期要继续学习C语言的数据结构，希望自己能够在攀登C语言大厦的过程中有更多的心得和体会，争取不再de一个bug在图书馆红温一个下午。

程序中出现的问题和改进

在写程序时我遇到最常见的问题便是和内存相关的问题，数组等依靠地址传递的变量越界问题，以及野指针问题等让人防不胜防，解决方法就是写好注释，这样在以后调试改代码时候就能节省再读一遍的时间。在以后工程肯定越来越大，多个函数甚至文件的综合处理，更需要对内存有清晰的理解。再写代码的过程中，有时候我还会遇到代码逻辑错误，导致输出的结果不符合预期，甚至十分滑稽让人摸不着头脑。还有遇到的问题就是对输入值检验不足，这个属于人机互动做的不够，在用户输入异常值时，程序应该给与提示，而我总忘记这点。

代码规范与调试技巧

在刚开始写代码时，我总觉得代码跑通就行，不注意格式与规范，等到现在再翻阅原来的代码，发现因为不注意代码规范，有的时候自己写的代码自己要理解半天，这时候才明白规范的重要。在调试时我会采用printf大法，检验程序是否进入某个部分，从而找出输出异常值的原因。也会采取间断点法，用软件自带的断点功能找出异常代码。前者主要用于可以编译但输出结果异常的情况，解决逻辑错误，而后者则多用于编译不通过时找出error所在。

学习记录和心得

学习C语言的过程是一段充满挑战与收获的旅程。最初，我被C语言简洁而强大的语法所吸引，它提供了丰富的数据类型和控制结构，既适合编写系统级软件，也适合开发高性能应用程序。然而，C语言的简洁也带来了挑战，尤其是内存管理方面，它要求程序员必须对内存管理有深刻的理解。指针作为C语言中最为独特且强大的特性之一，让我可以直接操作内存地址，为数据结构的实现提供了极大的灵活性，但同时也充满了风险，稍有不慎就可能导致内存泄漏、野指针或程序崩溃。我花费了大量的时间来理解指针的工作原理，并通过不断的练习来提高对指针操作的熟练度。随着对C语言的了解加深，我开始意识到编写可维护代码的重要性，学会了如何组织代码结构，使用函数和模块来提高代码的可读性和可维护性，也开始重视代码的注释和文档。调试是编程中不可或缺的一部分，我掌握了调试的技巧，通过设置断点等方法更有效地定位和解决问题。C语言的性能优化是一个复杂的话题，我学习了如何通过减少函数调用、优化数据结构和算法来提高程序的运行效率，了解到有时候为了性能，需要牺牲一些代码的可读性。总的来说，学习C语言让我掌握了一种强大的编程语言，也让我对计算机的工作原理有了更深入的理解，它的灵活性和高效性使其在许多领域都有着广泛的应用，而它的学习也让我为学习其他编程语言打下了坚实的基础。在未来的学习中，我希望能够将C语言的知识应用到实际的项目中，通过实践来进一步提升我的编程技能。

# 附录

完整实验代码附在此处

实验calendar01

#include <stdio.h>

// using namespace std;

int main(void)

{

    int month, day;

    puts("put input the date");

    scanf("%d %d", &month , &day);

    if (month==1)

    {

    }

    else if (month==2)

    {

        day=day+31;

    }

    else if (month==3)

    {

        day=day+60;

    }

    else if (month==4)

    {

        day=day+91;

    }

    else if (month==5)

    {

        day=day+121;

    }

    else if (month==6)

    {

        day=day+152;

    }

    else if (month==7)

    {

        day=day+182;

    }

    else if (month==8)

    {

        day=day+213;

    }

    else if (month==9)

    {

        day=day+244;

    }

    else if (month==10)

    {

        day=day+274;

    }

    else if (month==11)

    {

        day=day+305;

    }

    else if (month==12)

    {

        day=day+335;

    }

    day= day%7;

    if (day==0)

    {

        puts("Sunday");

    }

    if (day==1)

    {

        puts("Monday");

    }

    if (day==2)

    {

        puts("Tuesday");

    }

    if (day==3)

    {

        puts("Wednesday");

    }

    if (day==4)

    {

        puts("Thursday");

    }

    if (day==5)

    {

        puts("Friday");

    }

    if (day==6)

    {

        puts("Saturday");

    }

    return 0;

}

实验calendar02

#include<stdio.h>

int main()

{

    int month,mLength = 31,mStartDayOfWeek = 0;

    printf("Month Calendar of Year 2024, please input month(1-12):\n");

    int lengthJan = 31;

    int lengthFeb = lengthJan + 29;

    int lengthMar = lengthFeb + 31;

    int lengthApr = lengthMar + 30;

    int lengthMay = lengthApr + 31;

    int lengthJun = lengthMay + 30;

    int lengthJul = lengthJun + 31;

    int lengthAug = lengthJul + 31;

    int lengthSep = lengthAug + 30;

    int lengthOct = lengthSep + 31;

    int lengthNov = lengthOct + 30;

    int lengthDec = lengthNov + 31;

    scanf("%d",&month);

    if( month == 2 )

    {

        mStartDayOfWeek = lengthJan % 7;

        mLength = 29;

    }

    if( month == 3 )

    {

        mStartDayOfWeek = lengthFeb % 7;

        mLength = 31;

    }

    if( month == 4 )

    {

        mStartDayOfWeek = lengthMar % 7;

        mLength = 30;

    }

    if( month == 5 )

    {

        mStartDayOfWeek = lengthApr % 7;

        mLength = 31;

    }

    if( month == 6 )

    {

        mStartDayOfWeek = lengthMay % 7;

        mLength = 30;

    }

    if( month == 7 )

    {

        mStartDayOfWeek = lengthJun % 7;

        mLength = 31;

    }

    if( month == 8 )

    {

        mStartDayOfWeek = lengthJul % 7;

        mLength = 31;

    }

    if( month == 9 )

    {

        mStartDayOfWeek = lengthAug % 7;

        mLength = 30;

    }

    if( month == 10 )

    {

        mStartDayOfWeek = lengthSep % 7;

        mLength = 31;

    }

    if( month == 11 )

    {

        mStartDayOfWeek = lengthOct % 7;

        mLength = 30;

    }

    if( month == 12 )

    {

        mStartDayOfWeek = lengthNov % 7;

        mLength = 31;

    }

    printf("\n 2024-%d has total %d days",month,mLength);

    printf("\n 2024-%d-1 is No.%d day in that week.",month,mStartDayOfWeek + 1 );

    printf("\n\n");

    printf("%10s%10s%10s%10s%10s%10s%10s\n","Mon.","Tues.","Wed.","Thur.","Fri.","Sat.","Sun.");

    int i = 1;

    mStartDayOfWeek %= 7;

    while(i <= mStartDayOfWeek)

    {

        printf("%10s"," ");

        i++;

    }

    i = 1;

    while(i <= mLength){

        printf("%10d",i);

        i++;

        mStartDayOfWeek++;

        if(mStartDayOfWeek % 7 == 0)

        {

            mStartDayOfWeek %= 7;

            printf("\n");

        }

    }

    return 0;

}

实验calendar03

#include<stdio.h>

int main()

{

    printf("Week Calendar of Year 2024, please input week sequence(1-53):\n");

    int week,month,day,sum;

    scanf("%d",&week);

    sum = week \* 7 - 7;

    int lengthJan = 31;

    int lengthFeb = lengthJan + 29;

    int lengthMar = lengthFeb + 31;

    int lengthApr = lengthMar + 30;

    int lengthMay = lengthApr + 31;

    int lengthJun = lengthMay + 30;

    int lengthJul = lengthJun + 31;

    int lengthAug = lengthJul + 31;

    int lengthSep = lengthAug + 30;

    int lengthOct = lengthSep + 31;

    int lengthNov = lengthOct + 30;

    int lengthDec = lengthNov + 31;

    if(sum + 7 > 366)

    {

        printf("No.%d week in 2024 starts from No.%d day to No.%d day.\n\n",week,sum + 1,366);

    }else

    {

        printf("No.%d week in 2024 starts from No.%d day to No.%d day.\n\n",week,sum + 1,sum + 7);

    }

    printf("#W:%10s%10s%10s%10s%10s%10s%10s\n","Mon.","Tues.","Wed.","Thur.","Fri.","Sat.","Sun.");

    printf("%02d:",week);

    if(sum < lengthJan)

    {

        month = 1;

        int i = 1,day = sum;

        while(i <= 7)

        {

            sum++;

            i++;

            day++;

            if(sum == lengthJan + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }//January

    if(sum < lengthFeb)

    {

        month = 2;

        int i = 1,day = sum-lengthJan;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthFeb + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthMar)

    {

        month = 3;

        int i = 1,day = sum - lengthFeb;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthMar + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthApr)

    {

        month = 4;

        int i = 1,day = sum - lengthMar;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthJan + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthMay)

    {

        month = 5;

        int i = 1,day = sum - lengthApr;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthMay + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthJun)

    {

        month = 6;

        int i = 1,day = sum - lengthApr;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthJun + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthJul)

    {

        month = 7;

        int i = 1,day = sum - lengthJun;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthJul + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthAug)

    {

        month = 8;

        int i = 1,day = sum - lengthJul;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthAug + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthSep)

    {

        month = 9;

        int i = 1,day = sum - lengthAug;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthSep + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthOct)

    {

        month = 10;

        int i = 1,day = sum - lengthSep;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthOct + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthNov)

    {

        month = 11;

        int i = 1,day = sum - lengthOct;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthNov + 1)

            {

                day = 1;

                month++;

            }

            printf("%5s%02d.%02d","",month,day);

        }

        printf("\n");

        return 0;

    }

    if(sum < lengthDec)

    {

        month = 12;

        int i = 1,day = sum - lengthNov;

        while(i <= 7)

        {

            i++;

            sum++;

            day++;

            if(sum == lengthDec + 1)

            {

                return 0;

            }

            printf("%5s%02d.%02d","",month,day);

        }

    }

}

实验calendar04

#include<stdio.h>

//牛思途

int main()

{

    int month,referenceDay,day;

    int lengthJan = 31;

    int lengthFeb = lengthJan + 29;

    int lengthMar = lengthFeb + 31;

    int lengthApr = lengthMar + 30;

    int lengthMay = lengthApr + 31;

    int lengthJun = lengthMay + 30;

    printf("Spring Semester Calendar of Year 2024,please input reference day in Feb(1-29):\n");

    scanf("%d", &referenceDay);

    int startday = ( referenceDay + 2 ) % 7 + 1;

    referenceDay += ( startday == 1 ) ? 0 : ( 8 - startday );

    if( referenceDay <= 29 )

    {

        month = 2;

        day = referenceDay;

    }else

    {

        month = 3;

        day = referenceDay - 29;

    }

    referenceDay += lengthJan;

    int Islastweek = 0 , today\_week = 1 , week = 0;

    printf("%18s%s2024\n"," ","Spring Semester Calendar of Year");

    printf("#W:%10s%10s%10s%10s%10s%10s%10s\n","Mon.","Tues.","Wed.","Thur.","Fri.","Sat.","Sun.");

    while( Islastweek == 0 || today\_week != 1 )

    {

        if( today\_week == 1 )

        {

            printf("\n[%02d]",++week);

        }

        printf("%5s%02d.%02d","",month,day);

        day++;

        referenceDay++;

        today\_week = today\_week % 7 + 1;

        if(month == 7)

        {

            Islastweek = 1;

        }

        if( month == 2 && referenceDay > lengthFeb )

        {

            month++;

            day -= lengthFeb - lengthJan;

        }

        if( month == 3 && referenceDay > lengthMar )

        {

            month++;

            day -= lengthMar - lengthFeb;

        }

        if( month == 4 && referenceDay > lengthApr )

        {

            month++;

            day -= lengthApr - lengthMar;

        }

        if( month == 5 && referenceDay > lengthMay )

        {

            month++;

            day -= lengthMay - lengthApr;

        }

        if( month == 6 && referenceDay > lengthJun )

        {

            month++;

            day -= lengthJun - lengthMay;

        }

    }

    return 0;

}

实验calendar05

#include <stdio.h>

//牛思途

int getNextMonday(int daySeq);

int getThisSunday(int daySeq);

int getDaySeq(int month, int day);

void printOneDay(int daySeq);

int main() {

    int referenceDay;

    printf("[Spring Semester Calendar of Year 2021, please input reference day in Feb(1-28): ");

    scanf("%d", &referenceDay);

    int sStartSeqOfYear = getNextMonday(getDaySeq(2, referenceDay));

    int sEndSeqOfYear = getThisSunday(getDaySeq(7, 1));

    int daySeqOfYear, daySeqOfWeek, weekSeqOfSemester;

    for (daySeqOfYear = sStartSeqOfYear, daySeqOfWeek = 0, weekSeqOfSemester = 1;

         daySeqOfYear <= sEndSeqOfYear;

         daySeqOfYear++, daySeqOfWeek++, daySeqOfWeek %= 7) {

        if (daySeqOfWeek == 0) {

            printf("[%02d]", weekSeqOfSemester);

        }

        printOneDay(daySeqOfYear);

        if (daySeqOfWeek == 6) {

            printf("\n");

            weekSeqOfSemester++;

        }

    }

    return 0;

}

int getNextMonday(int daySeq) {

    return daySeq + 7;

}

int getThisSunday(int daySeq) {

    return daySeq;

}

int getDaySeq(int month, int day) {

    int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    int seq = 0;

    for (int i = 0; i < month - 1; i++) {

        seq += daysInMonth[i];

    }

    seq += day;

    return seq;

}

void printOneDay(int daySeq) {

    printf("%2d ", daySeq % 30 + 1);

}

实验calendar06

#include <stdio.h>

//牛思途

int isLeapYear(int year);

int getDaySeqOnJan1(int year);

int getMonthLength(int year, int month);

int getDaySeq(int year, int month, int day);

int getMonth(int year, int daySeqOfYear);

int getDay(int year, int daySeqOfYear);

int getDaySeqOfWeek(int year, int daySeqOfYear);

int getNextMonday(int year, int daySeqOfYear);

int getThisMonday(int year, int daySeqOfYear);

int getThisSunday(int year, int daySeqOfYear);

void printOneDay(int year, int daySeqOfYear);

int main() {

    int currentYear = 2021;

    int referenceDay;

    printf("Please input reference day in Next Jan(1-31): ");

    scanf("%d", &referenceDay);

    int sStartSeqOfYear = getThisMonday(currentYear, getDaySeq(currentYear, 9, 1));

    int sEndSeqOfYear = getThisSunday(currentYear + 1, getDaySeq(currentYear + 1, 1, referenceDay));

    int daySeqOfYear, daySeqOfWeek, weekSeqOfSemester;

    int currentYearLength = getMonthLength(currentYear, 12);

    for (daySeqOfYear = sStartSeqOfYear, daySeqOfWeek = 0, weekSeqOfSemester = 1;

         daySeqOfYear <= sEndSeqOfYear;

         daySeqOfYear++, daySeqOfWeek++, daySeqOfWeek %= 7) {

        if (daySeqOfWeek == 0) {

            printf("[%02d]", weekSeqOfSemester);

        }

        if (daySeqOfYear <= currentYearLength) {

            printOneDay(currentYear, daySeqOfYear);

        } else {

            printOneDay(currentYear + 1, daySeqOfYear - currentYearLength);

        }

        if (daySeqOfWeek == 6) {

            printf("\n");

            weekSeqOfSemester++;

        }

    }

    return 0;

}

int isLeapYear(int year) {

    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

}

int getDaySeqOnJan1(int year) {

    int seq = 0;

    for (int i = 1; i < year; i++) {

        if (isLeapYear(i)) {

            seq += 366;

        } else {

            seq += 365;

        }

    }

    return seq;

}

int getMonthLength(int year, int month) {1

    int lengths[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    if (month == 2 && isLeapYear(year)) {

        return 29;

    }

    return lengths[month - 1];

}

int getDaySeq(int year, int month, int day) {

    int seq = getDaySeqOnJan1(year);

    for (int i = 1; i < month; i++) {

        seq += getMonthLength(year, i);

    }

    seq += day - 1;

    return seq;

}

int getMonth(int year, int daySeqOfYear) {

    int seq = 0;

    for (int i = 1; i <= 12; i++) {

        seq += getMonthLength(year, i);

        if (seq > daySeqOfYear) {

            return i;

        }

    }

    return 1;

}

int getDay(int year, int daySeqOfYear) {

    int seq = 0;

    for (int i = 1; i <= 12; i++) {

        seq += getMonthLength(year, i);

        if (seq > daySeqOfYear) {

            return daySeqOfYear - (seq - getMonthLength(year, i)) + 1;

        }

    }

    return 1;

}

int getDaySeqOfWeek(int year, int daySeqOfYear) {

    return (daySeqOfYear - getDaySeqOnJan1(year) + getDaySeqOnJan1(year) % 7) % 7;

}

int getNextMonday(int year, int daySeqOfYear) {

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    return daySeqOfYear + (8 - dayOfWeek);

}

int getThisMonday(int year, int daySeqOfYear) {

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    return daySeqOfYear - dayOfWeek;

}

int getThisSunday(int year, int daySeqOfYear) {

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    return daySeqOfYear + (7 - dayOfWeek);

}

void printOneDay(int year, int daySeqOfYear) {

    int month = getMonth(year, daySeqOfYear);

    int day = getDay(year, daySeqOfYear);

    if (day == 1) {

        printf("%02d.%02d ", month, day);

    } else {

        printf("%2d ", day);

    }

}

1. 实验calenda08 月历数组
2. main.c
3. #include <stdio.h>
4. //牛牛
5. #define YEAR\_MIN 2000
6. #define YEAR\_MAX 2025
7. #define YEAR\_NUM 5
8. const int Month\_NORMAL\_YEAR[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
9. const int Month\_LEAP\_YEAR[12] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
10. int isLeapYear(int year);
11. int getDaySeqOnJan1(int year);
12. int getDaySeq(int year, int month, int day);
13. int getWeekSeqOfYear(int year, int month, int day);
14. int getMonth(int year, int daySeqOfYear);
15. int getDay(int year, int daySeqOfYear);
16. int getDaySeqOfWeek(int year, int daySeqOfYear);
17. int getThisMonday(int year, int daySeqOfYear);
18. int getThisSunday(int year, int daySeqOfYear);
19. void printOneDay(int year, int daySeqOfYear, int formatType);
20. void printOneWeek(int year, int weekSeqOfYear, int weekSeqShow);
21. void setYearArray(int yearArray[], int yearNum, int yearStart);
22. int main() {
23. int inputYear, Years[YEAR\_NUM];
24. printf("Spring Calendar Comparison in %d years, please input the first year (%d-%d):\n", YEAR\_NUM, YEAR\_MIN, YEAR\_MAX - YEAR\_NUM);
25. scanf("%d", &inputYear);
26. if (inputYear < YEAR\_MIN || inputYear + YEAR\_NUM > YEAR\_MAX) {
27. printf("Sorry, the input year is not supported.\n");
28. return 1;
29. }
30. setYearArray(Years, YEAR\_NUM, inputYear);
31. int i;
32. int startDays[YEAR\_NUM] = {0};
33. for (i = 0; i < YEAR\_NUM; i++) {
34. printf("\n%s%s%d\n", "First week in spring Calendar of Year ", Years[i]);
35. printf("\t%10s%10s%10s%10s%10s%10s%10s\n", "Mon.", "Tues.", "Wed.", "Thur.", "Fri.", "Sat.", "Sun.");
36. printOneWeek(Years[i], getWeekSeqOfYear(Years[i], 9, 4), 1);
37. startDays[i] = getThisMonday(Years[i], getDaySeq(Years[i], 9, 4));
38. }
39. int min = 0;
40. for (i = 1; i < YEAR\_NUM; i++) {
41. if (startDays[i] < startDays[min]) {
42. min = i;
43. }
44. }
45. printf("\n%s%s%d\n", "Earliest Spring Semester is in Year ", Years[min]);
46. return 0;
47. }
48. int isLeapYear(int year) {
49. return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
50. }
51. int getDaySeqOnJan1(int year) {
52. int daySeq = 0;
53. for (int y = 1; y < year; y++) {
54. if (isLeapYear(y)) {
55. daySeq += 2;
56. }
57. daySeq = (daySeq + 365) % 7;
58. }
59. return daySeq;
60. }
61. int getDaySeq(int year, int month, int day) {
62. int daySeq = 0;
63. if (isLeapYear(year)) {
64. for (int m = 1; m < month; m++) {
65. daySeq += Month\_LEAP\_YEAR[m - 1];
66. }
67. } else {
68. for (int m = 1; m < month; m++) {
69. daySeq += Month\_NORMAL\_YEAR[m - 1];
70. }
71. }
72. daySeq += day;
73. return daySeq;
74. }
75. int getWeekSeqOfYear(int year, int month, int day) {
76. int daySeq = getDaySeq(year, month, day);
77. int weekSeq = (daySeq + getDaySeqOnJan1(year)) / 7;
78. return weekSeq;
79. }
80. int getMonth(int year, int daySeqOfYear) {
81. int month = 1;
82. if (isLeapYear(year)) {
83. while (daySeqOfYear > Month\_LEAP\_YEAR[month - 1]) {
84. daySeqOfYear -= Month\_LEAP\_YEAR[month - 1];
85. month++;
86. }
87. } else {
88. while (daySeqOfYear > Month\_NORMAL\_YEAR[month - 1]) {
89. daySeqOfYear -= Month\_NORMAL\_YEAR[month - 1];
90. month++;
91. }
92. }
93. return month;
94. }
95. int getDay(int year, int daySeqOfYear) {
96. int month = getMonth(year, daySeqOfYear);
97. int day = daySeqOfYear - (isLeapYear(year) ? Month\_LEAP\_YEAR[month - 2] : Month\_NORMAL\_YEAR[month - 2]);
98. return day;
99. }
100. int getDaySeqOfWeek(int year, int daySeqOfYear) {
101. int daySeq = (daySeqOfYear + getDaySeqOnJan1(year)) % 7;
102. return daySeq;
103. }
104. int getThisMonday(int year, int daySeqOfYear) {
105. int daySeq = getDaySeqOfWeek(year, daySeqOfYear);
106. return (daySeq + 6) % 7;
107. }
108. int getThisSunday(int year, int daySeqOfYear) {
109. int daySeq = getDaySeqOfWeek(year, daySeqOfYear);
110. return daySeq;
111. }
112. void printOneDay(int year, int daySeqOfYear, int formatType) {
113. int month = getMonth(year, daySeqOfYear);
114. int day = getDay(year, daySeqOfYear);
115. int daySeq = getDaySeqOfWeek(year, daySeqOfYear);
116. printf("%d-%02d-%02d %s\n", year, month, day, (formatType == 1) ? "Mon" : "Sun");
117. }
118. void printOneWeek(int year, int weekSeqOfYear, int weekSeqShow) {
119. int startDay = (weekSeqShow - 1) \* 7;
120. int daySeq = startDay;
121. for (int i = 0; i < 7; i++) {
122. int month = getMonth(year, daySeq);
123. int day = getDay(year, daySeq);
124. printf("%02d ", day);
125. daySeq++;
126. }
127. printf("\n");
128. }
129. void setYearArray(int yearArray[], int yearNum, int yearStart) {
130. for (int i = 0; i < yearNum; i++) {
131. yearArray[i] = yearStart + i;
132. }
133. }

s

1. 实验calendar09

（1）main.c

#include <stdio.h>

#include <stdlib.h>

int isLeapYear(int year);

int getDaySeqOnJan1(int year);

int getMonthLength(int year, int month);

int getDaySeq(int year, int month, int day);

int getWeekSeqOfYear(int year, int month, int day);

int getMonth(int year, int daySeqOfYear);

int getDay(int year, int daySeqOfYear);

int getDaySeqOfWeek(int year, int daySeqOfYear);

int getNextMonday(int year, int daySeqOfYear);

int getThisMonday(int year, int daySeqOfYear);

int getThisSunday(int year, int daySeqOfYear);

void printOneDay(int year, int daySeqOfYear);

void printOneWeek(int year, int weekSeqOfYear, int weekSeqShow);

void setYearArray(int yearArray[], int yearNum, int yearStart);

void initialDays(int Years[], int Days[][366][4], int yearNum);

#define YEAR\_NUM 6

#define YEAR\_MIN 2020

#define YEAR\_MAX 2025

int main() {

    int inputMonth, inputDay, Years[YEAR\_NUM];

    int Days[YEAR\_NUM][366][4] = {0};

    printf("Finding Birthday in year (%d-%d), please input the month and day:\n", YEAR\_MIN, YEAR\_MAX);

    scanf("%d %d", &inputMonth, &inputDay);

    if (inputMonth < 1 || inputMonth > 12 || inputDay < 1 || inputDay > getMonthLength(YEAR\_MIN, inputMonth)) {

        printf("Sorry, the input month and day are invalid.\n");

        return 1;

    }

    setYearArray(Years, YEAR\_NUM, YEAR\_MIN);

    initialDays(Years, Days, YEAR\_NUM);

    int totalNum = 0, weekendNum = 0;

int i;

for (i = 0; i < YEAR\_NUM; i++) {

    int year = Years[i];

    int daySeqOfYear = getDaySeq(year, inputMonth, inputDay);

    if (daySeqOfYear <= (isLeapYear(year) ? 366 : 365)) {

        printf("\nBirthday in Year %d\n", year);

        if (!(isLeapYear(year)) && inputMonth == 2 && inputDay == 29) {

            printf("Not found.\n");

        } else {

            int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

            if (dayOfWeek == 6 || dayOfWeek == 7) {

                weekendNum++;

            }

            totalNum++;

            printf("#M:%10s%10s%10s%10s%10s%10s%10s\n", "Mon.", "Tues.", "Wed.", "Thur.", "Fri.", "Sat.", "Sun.");

            printOneWeek(year, getWeekSeqOfYear(year, inputMonth, inputDay), 1);

        }

    }

}

printf("\nTotal %d birthdays are found, %d of them are in weekends.\n", totalNum, weekendNum);

    return 0;

}

int isLeapYear(int year) {

    int i;

    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {

        i = 1;

    } else {

        i = 0;

    }

    return i;

}

int getDaySeqOnJan1(int year) {

    int days = 0;

    int i;

    for (int i = 1990; i < year; i++) {

        days += isLeapYear(i) ? 366 : 365;

    }

    return days % 7;

}

int getMonthLength(int year, int month) {

    int daysInMonth[] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    if (isLeapYear(year)) {

        daysInMonth[1] = 29;

    } else {

        daysInMonth[1] = 28;

    }

    return daysInMonth[month - 1];

}

int getDaySeq(int year, int month, int day) {

    int totalDays = 0;

    int i;

    for (i = 1; i < month; i++) {

        totalDays += getMonthLength(year, i);

    }

    totalDays += day;

    return totalDays;

}

int getDaySeqOfYear(int year, int month, int day) {

    int daySeqOfYear = getDaySeq(year, month, day);

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    int weekSeqOfYear = (daySeqOfYear + dayOfWeek - 1) % 7 + 1;

    return weekSeqOfYear;

}

int getMonth(int year, int daySeqOfYear) {

    int month = 1;

    while (month <=12 && daySeqOfYear > getMonthLength(year, month)) {

        daySeqOfYear -= getMonthLength(year, month);

        month++;

    }

    return month;

}

int getDay(int year, int daySeqOfYear) {

    int month = 1;

    while (month <=12 && daySeqOfYear > getMonthLength(year, month)) {

        daySeqOfYear -= getMonthLength(year, month);

        month++;

    }

    return daySeqOfYear;

}

int getDaySeqOfWeek(int year, int daySeqOfYear) {

    int dayOfWeek = getDaySeqOnJan1(year);

    int daySeqOfWeek = (dayOfWeek + daySeqOfYear - 1) % 7 + 1;

    return daySeqOfWeek;

}

int getNextMonday(int year, int daySeqOfYear) {

    int nextMonday;

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    if (dayOfWeek == 1) {

        nextMonday = daySeqOfYear;

    } else {

        nextMonday = daySeqOfYear + (8 - dayOfWeek);

    }

    return nextMonday;

}

int getThisMonday(int year, int daySeqOfYear) {

    int thisMonday;

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    if (dayOfWeek == 1) {

        thisMonday = daySeqOfYear;

    } else {

        thisMonday = daySeqOfYear - (dayOfWeek - 1);

    }

    return thisMonday;

}

int getThisSunday(int year, int daySeqOfYear) {

    int thisSunday;

    int dayOfWeek = getDaySeqOfWeek(year, daySeqOfYear);

    if (dayOfWeek == 7) {

        thisSunday = daySeqOfYear;

    } else {

        thisSunday = daySeqOfYear + (7 - dayOfWeek);

    }

    return thisSunday;

}

void printOneDay(int year, int daySeqOfYear) {

    int month = getMonth(year, daySeqOfYear);

    int day = getDay(year, daySeqOfYear);

    if (month == 13) {

        month = 1;

    }

    printf(" %02d.%02d ", month, day);

}

void printOneWeek(int year, int weekSeqOfYear, int weekSeqShow) {

    int startDaySeqOfYear;

    if (getThisMonday(year, 1) == 1) {

        startDaySeqOfYear = (weekSeqOfYear - 1) \* 7 + getThisMonday(year, 1);

    } else {

        startDaySeqOfYear = (weekSeqOfYear - 1) \* 7 + getThisMonday(year, 1) + 7;

    }

    int endDaySeqOfYear = startDaySeqOfYear + 6;

    if (weekSeqShow == 1) {

        printf("[%02d] ", weekSeqOfYear);

    }

    for (int daySeqOfYear = startDaySeqOfYear; daySeqOfYear <= endDaySeqOfYear; daySeqOfYear++) {

        printOneDay(year, daySeqOfYear);

    }

    printf("\n");

}

void setYearArray(int yearArray[], int yearNum, int yearStart) {

    int i;

    for (i = 0; i < yearNum; i++) {

        yearArray[i] = yearStart + i;

    }

}

void initialDays(int Years[], int Days[][366][4], int yearNum) {

    int year, month, day, yearLength, weekSeq, seqOfWeek;

    int i, j;

    for (i = 0; i < yearNum; i++) {

        year = Years[i];

        yearLength = isLeapYear(year) ? 366 : 365;

        for (j = 0; j < yearLength; j++) {

            Days[i][j][0] = getMonth(year, j + 1);

            Days[i][j][1] = getDay(year, j + 1);

            Days[i][j][2] = getWeekSeqOfYear(year, Days[i][j][0], Days[i][j][1]);

            Days[i][j][3] = getDaySeqOfWeek(year, j + 1);

        }

    }

}

三、实验calendar10

（1）main.c

#include<stdio.h>

#include<math.h>

#define YEAR\_NUM 9

#define YEAR\_MIN 2020

#define YEAR\_MAX 2028

#define DATE\_INFO\_BRIEF 1

#define DATE\_INFO\_FULL 2

#define DATE\_NOSHOW 3

#define DATE\_STAR 4

typedef struct main

{

int year;

int month;

int day;

int daySeq;

int weekSeq;

int weekDay;/\* data \*/

} Day;

const int Month\_NORMAL\_YEAR[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

const int Month\_LEAP\_YEAR[12] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

int isLeapYear(int year);

int getDaySeqOnJan1(int year);

// Functions about month and day

int getMonthLength(int year, int month);

int getDaySeq(int year, int month, int day);

int getWeekSeqOfYear(int year, int month, int day);

// Functions for properties of one day

int getMonth(int year, int daySeqOfYear);

int getDay(int year, int daySeqOfYear);

int getDaySeqOfWeek(int year, int daySeqOfYear);

void setYearArray(int Years[], int yearNum, int yearMin);

int isDay(int year, int month, int day);

Day setDay(int year, int month, int day);

Day getDayBefore(Day currentday, int interval);

Day getDayAfter(Day currentday, int interval);

int getTwoDaysInterval(Day startDay, Day endDay);

void printDay(Day currentday, int displayFormat);

int main(void)

{

int i, inputMonth, inputDay, displayFormat;

Day birthday, prepareday,printStartDay,currentDay;

int printDayRange = 3;

int Years[YEAR\_NUM];

setYearArray(Years, YEAR\_NUM, YEAR\_MIN);

printf("Finding Birthday in year (%d-%d), please input the month and day : ", YEAR\_MIN, YEAR\_MAX);

scanf("%d%d", &inputMonth, &inputDay);

if (inputMonth < 1 || inputMonth > 12 || inputDay < 1 || inputDay > 31)

{

printf("Sorry, the input month and day are invalid.\n");

return 1;

}

for ( i = 0 ; i < YEAR\_NUM ; i ++ )

{

printf( "\n%s%s%d\n", " ", "Birthday in Year ", Years[i] );

if ( isDay(Years[i], inputMonth, inputDay) == 1 )

{

printf(" Not found.\n");

continue;

}

birthday = setDay ( Years[i], inputMonth, inputDay );

prepareday = getDayBefore( birthday, printDayRange );

printStartDay = getDayBefore( prepareday, prepareday.weekDay );

printf(" #W:%10s%10s%10s%10s%10s%10s%10s\n[%02d]","Mon.","Tues.","Wed.","Thur.","Fri.","Sat.","Sun.",printStartDay.weekSeq);

for ( currentDay = printStartDay; getTwoDaysInterval(currentDay, birthday) <= 0;

currentDay = getDayAfter( currentDay, 1 ) )

{

if ( getTwoDaysInterval(currentDay, prepareday) < 0 )

{

displayFormat = DATE\_NOSHOW;

}

else if ( getTwoDaysInterval(currentDay, prepareday) == 0 )

{

displayFormat = DATE\_INFO\_FULL;

}

else if ( getTwoDaysInterval(currentDay, birthday) < 0 )

{

displayFormat = DATE\_INFO\_BRIEF;

}

else if ( getTwoDaysInterval(currentDay, birthday) == 0 )

{

displayFormat = DATE\_STAR;

}

printDay(currentDay, displayFormat);

if(currentDay.weekDay == 6 && displayFormat != DATE\_STAR)

{

printf("\n #W:%10s%10s%10s%10s%10s%10s%10s\n[%02d]","Mon.","Tues.","Wed.","Thur.","Fri.","Sat.","Sun.", birthday.weekSeq );

}

}

printf("\n");

}

}

int isLeapYear(int year)

{

if (year % 4 == 0 && year % 100 != 0)

{

return 366;

}

else

{

return 365;

}

}

int getDaySeqOnJan1(int year)

{

int day = 1;

while (year > 2024)

{

day += isLeapYear(year - 1);

year --;

}

while(year < 2024)

{

day -= isLeapYear(year);

year ++;

}

if (day > 0)

{

day = (day - 1) % 7;

}

else

{

day = (day - 1) % 7 + 7;

}

return day;

}

int getMonthLength(int year, int month)

{

int day;

switch(month)

{

case 1:

day = 31;

break;

case 2:

if (year % 4 == 0 && year % 100 != 0)

{

day = 29;

}

else

{

day = 28;

}

break;

case 3:

day = 31;

break;

case 4:

day = 30;

break;

case 5:

day = 31;

break;

case 6:

day = 30;

break;

case 7:

day = 31;

break;

case 8:

day = 31;

break;

case 9 :

day = 30;

break;

case 10:

day = 31;

break;

case 11:

day = 30;

break;

case 12:

day = 31;

break;

}

return day;

}

int getDaySeq(int year, int month, int day)

{

/\*for(;month > 1 ;month --)

{

day += getMonthLength(year, month);

}\*/

if(isLeapYear(year) == 365)

{

for(;month > 1; month --)

{

day += Month\_NORMAL\_YEAR[month - 2];

}

}

else

{

for(;month > 1; month --)

{

day += Month\_LEAP\_YEAR[month - 2];

}

}

return day;

}

int getWeekSeqOfYear(int year, int month, int day)

{

int daySeqOfYear = getDaySeq(year, month, day);

daySeqOfYear = daySeqOfYear + getDaySeqOnJan1(year) - 1;

int week;

week = daySeqOfYear / 7 + 1;

return week;

}

int getMonth(int year, int daySeqOfYear)

{

int month ;

for(month = 1; daySeqOfYear > getMonthLength(year, month); month ++)

{

daySeqOfYear -= getMonthLength(year, month);

}

return month;

}

int getDay(int year, int daySeqOfYear)

{

int month ;

for(month = 1;daySeqOfYear > getMonthLength(year, month); month ++)

{

daySeqOfYear -= getMonthLength(year, month);

}

return daySeqOfYear;

}

int getDaySeqOfWeek(int year, int daySeqOfYear)

{

int day = (daySeqOfYear + getDaySeqOnJan1(year) - 2) % 7;

return day;

}

void setYearArray(int Years[], int yearNum, int yearMin)

{

int i = 0;

for(i = 0; i < yearNum; i++)

{

Years[i] = yearMin + i;

}

return ;

}

int isDay(int year, int month, int day)

{

if(isLeapYear(year) != 366 && month == 2 && day == 29)

{

return 1;

}

return 0;

}

Day setDay(int year, int month, int day)

{

Day d;

d.year = year;

d.month = month;

d.day = day;

d.daySeq = getDaySeq(year, month, day);

d.weekSeq = getWeekSeqOfYear(year, month, day);

d.weekDay = getDaySeqOfWeek(year, d.daySeq);

return d;

}

Day getDayBefore(Day currentday, int interval)

{

/\*if (interval > 28)

{

printf("Input error in function getDayBefore!");

return currentday;

}

currentday.day -= interval;

currentday.weekDay -= interval;

currentday.daySeq -= interval;

if(currentday.weekDay <= 6)

{

currentday.weekSeq -= (currentday.weekDay / 7 + 1);

currentday.weekDay = (currentday.weekDay % 7 + 7 ) % 7;

}

if(currentday.day < 1)

{

if(currentday.month == 1)

{

currentday.year --;

currentday.month = 12;

currentday.daySeq += isLeapYear(currentday.year);

currentday.day += 31;

currentday.weekSeq = getWeekSeqOfYear(currentday.year, currentday.month, currentday.day);

}

else

{

currentday.month --;

currentday.day += getMonthLength(currentday.year, currentday.month);

}

}\*/

currentday.daySeq -= interval;

while(currentday.daySeq <= 0)

{

currentday.year --;

currentday.daySeq += isLeapYear(currentday.year );

}

currentday.month = getMonth(currentday.year, currentday.daySeq);

currentday.day = getDay(currentday.year,currentday.daySeq);

currentday.weekDay = getDaySeqOfWeek(currentday.year, currentday.daySeq);

currentday.weekSeq = getWeekSeqOfYear(currentday.year, currentday.month, currentday.day);

return currentday;

}

Day getDayAfter(Day currentday, int interval)

{

currentday.daySeq += interval;

while(currentday.daySeq > isLeapYear(currentday.year))

{

currentday.year ++;

currentday.daySeq -= isLeapYear(currentday.year - 1);

}

currentday.month = getMonth(currentday.year, currentday.daySeq);

currentday.day = getDay(currentday.year,currentday.daySeq);

currentday.weekDay = getDaySeqOfWeek(currentday.year, currentday.daySeq);

currentday.weekSeq = getWeekSeqOfYear(currentday.year, currentday.month, currentday.day);

return currentday;

}

int getTwoDaysInterval(Day startDay, Day endDay)

{

int interval = 0;

while(endDay.year != startDay.year)

{

if(endDay.year > startDay.year)

{

interval -= isLeapYear(startDay.year);

startDay.year ++;

}

else

{

interval += isLeapYear(startDay.year - 1);

startDay.year --;

}

}

interval += (startDay.daySeq - endDay.daySeq);

return interval;

}

void printDay(Day currentday, int displayFormat)

{

switch (displayFormat)

{

case 1:

printf("%10d", currentday.day);

break;

case 2:

printf("%4d.%02d.%02d", (currentday.year % 100), currentday.month, currentday.day);

break;

case 3:

printf("%10s", "");

break;

case 4:

printf(" %02d.%02d\*", currentday.month, currentday.day);

break;

}

return;

}