

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 | 肖士博 | 学号 | U202411347 |

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

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

# 实验目的

完成日历系列代码（日历系列、大数计算系列，选择其中之一）。

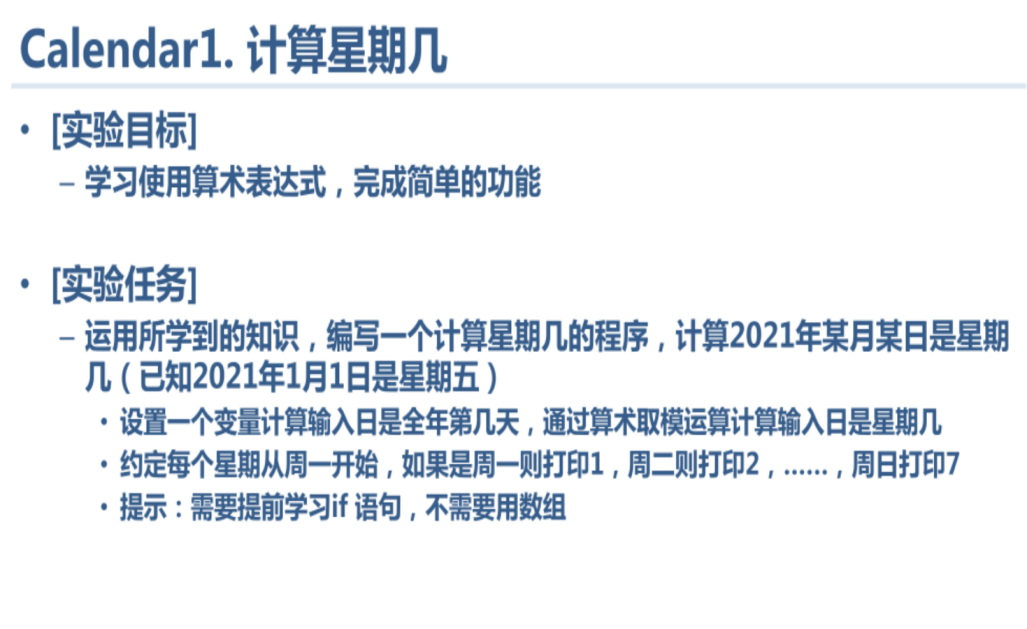
# 实验环境

操作系统：Windows 10

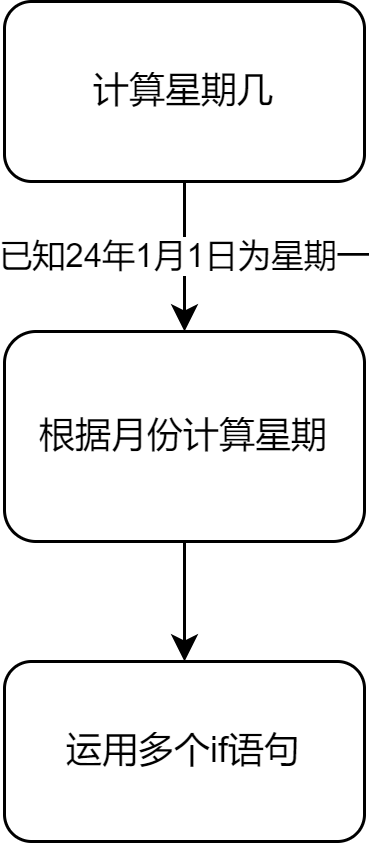
编程工具：CodeBlocks 16.01

# 实验一

## 实验任务



## 实验步骤



## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 是否正常输出是星期几 | 随机输入日期与实际对比 |  |

## 实验结论

达到功能目标

## 实验总结

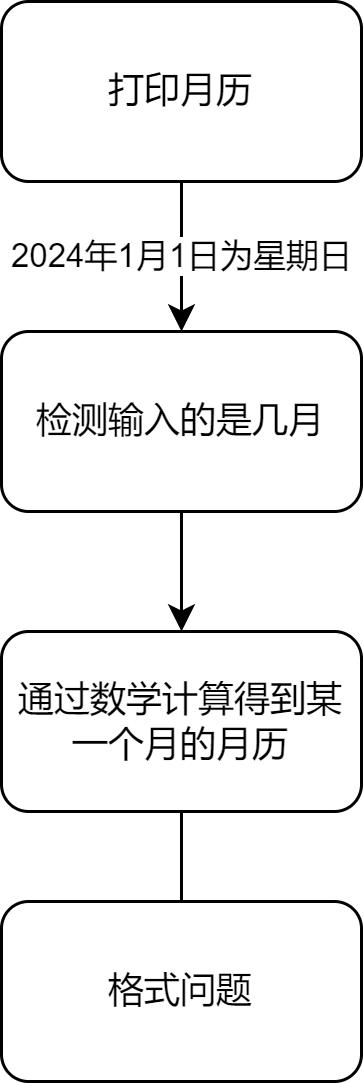
可使用更简单的方式优化代码，如合并一些相同逻辑的代码

# 实验二

## 实验任务



## 实验步骤



## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方式 | 测试结果 |
| 1 | 格式 | 输入 |  |
| 2 | 是否检查输入正确与否 | 输入 |  |
| 3 | 是否正确输出日期 | 输入 |  |

## 实验结论

代码达到功能目标

## 实验总结

可以简化，同实验一

# 实验三

## 实验任务



## 实验步骤

## 打印周历

## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 是否有提示信息 | 输入 |  |
| 2 | 是否有报错 | 输入 |  |
| 3 | 输出日期是否正确 | 输入 |  |

## 实验结论

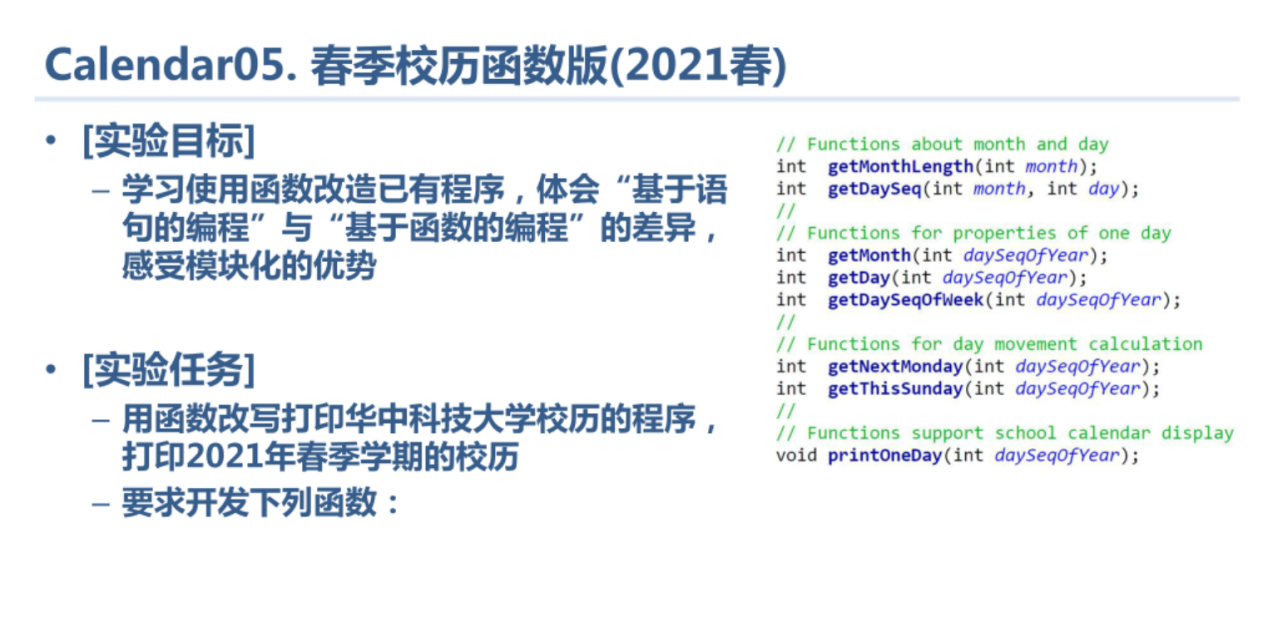
代码达到功能目标

## 实验总结

无

# 实验四

## 实验任务



## 实验步骤

## 春季

## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 是否正确报错 | 输入错误数字 |  |
| 2 | 是否是从下一周开始 | 输入 |  |
| 3 | 是否有感叹号与正确月份表示 | 输入 |  |

## 实验结论

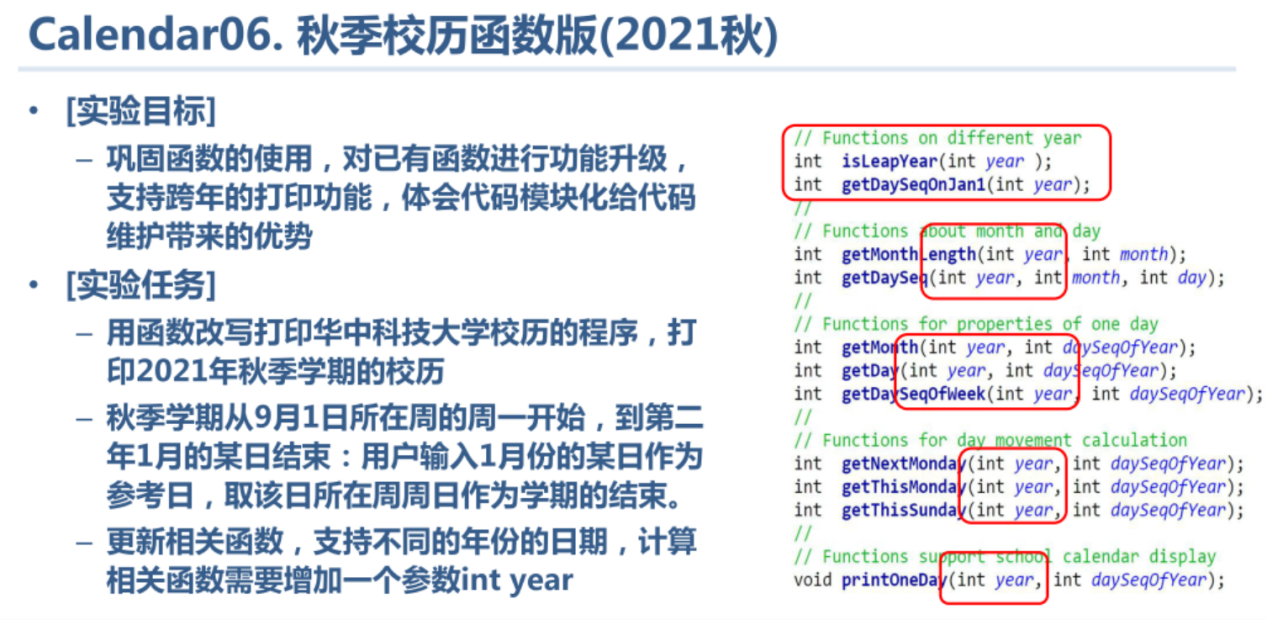
代码达到功能目标

## 实验总结

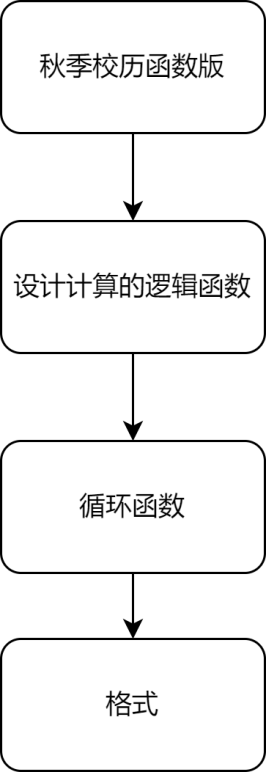
无

# 实验五

## 实验任务



## 实验步骤



## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 对应日期是否正确 | 输入某一年 |  |
| 2 | 是否在月初输出月 | 输入 |  |
| 3 | 是否有  ！ | 输入 |  |

## 实验结论

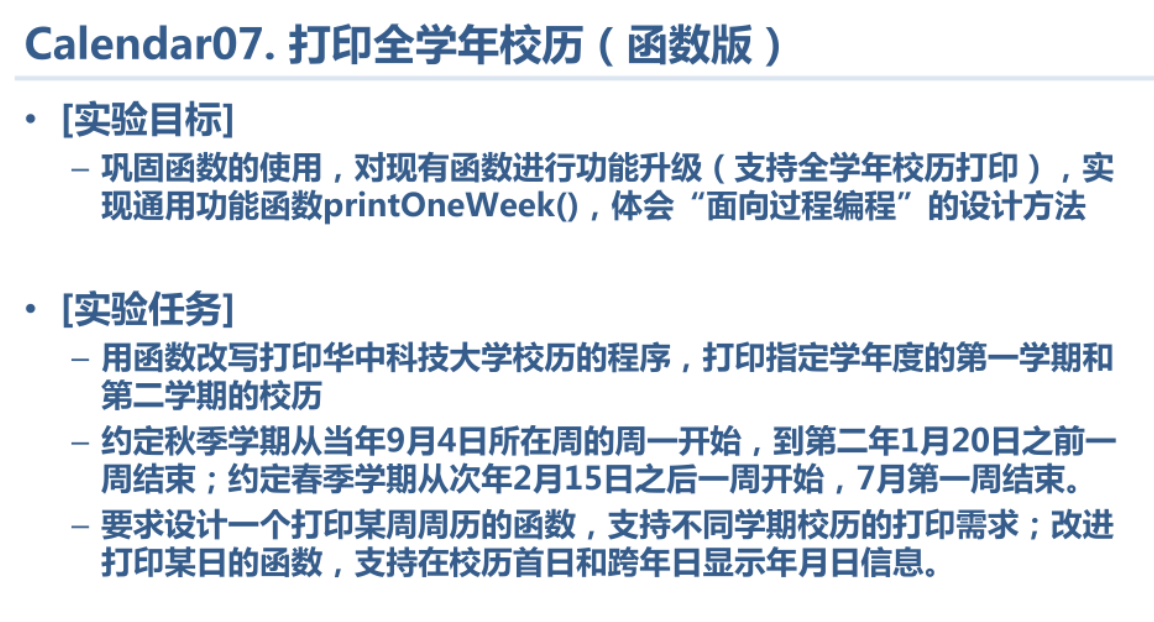
代码达到功能目标

## 实验总结

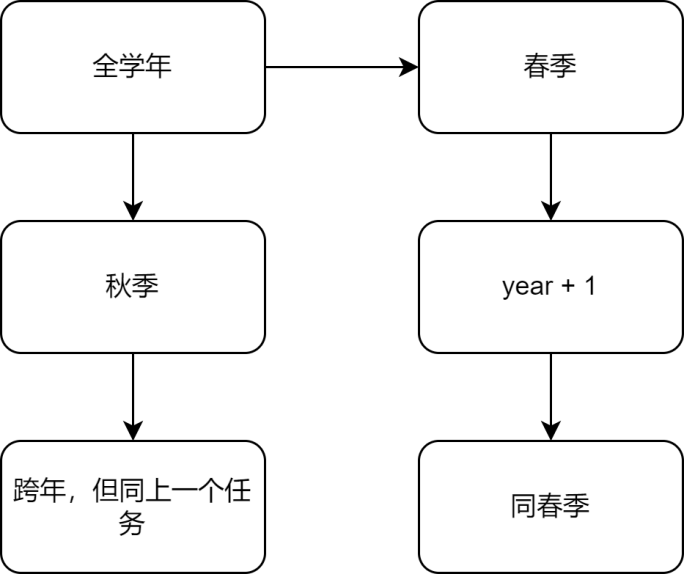
可优化getDaySeq函数

# 实验六

## 实验任务



## 实验步骤



## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 是否报错 | 输入范围之外的数 |  |
| 2 | 秋季 |  |  |
| 3 | 春季 |  |  |

## 实验结论

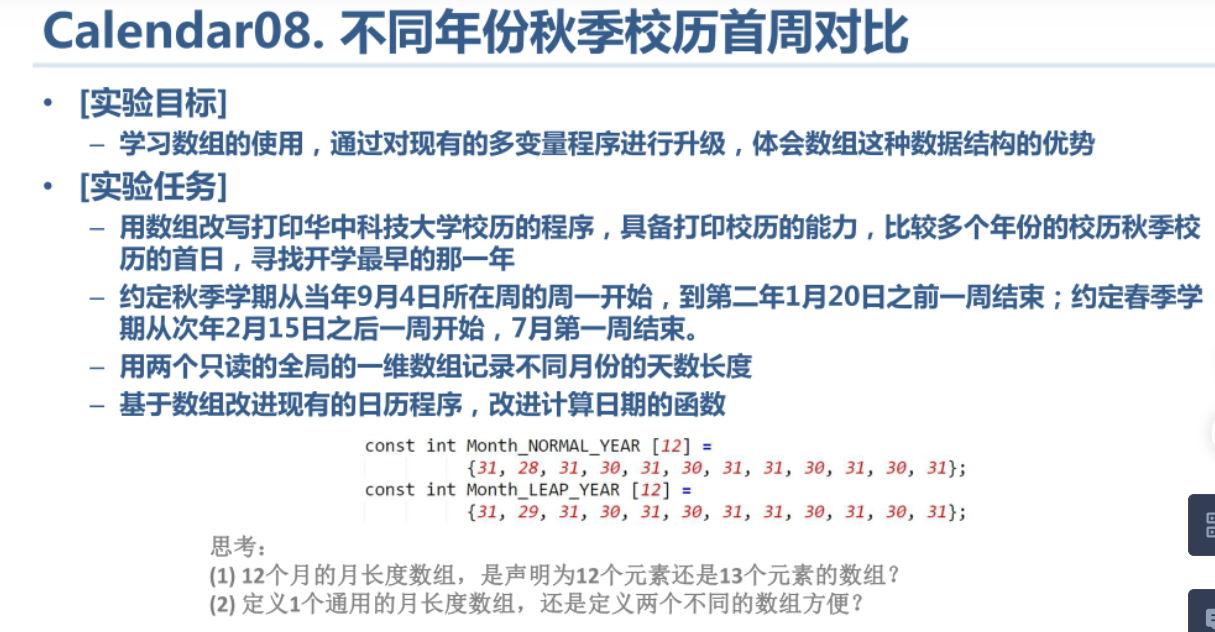
代码达到功能目标

## 实验总结

无

# 实验七

## 实验任务



## 实验步骤

同上一个实验

## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 是否 为最早 |  |  |
| 2 | 报错 |  |  |
| 3 | 日期是否正确 |  |  |

## 实验结论

代码达到功能目标

## 实验总结

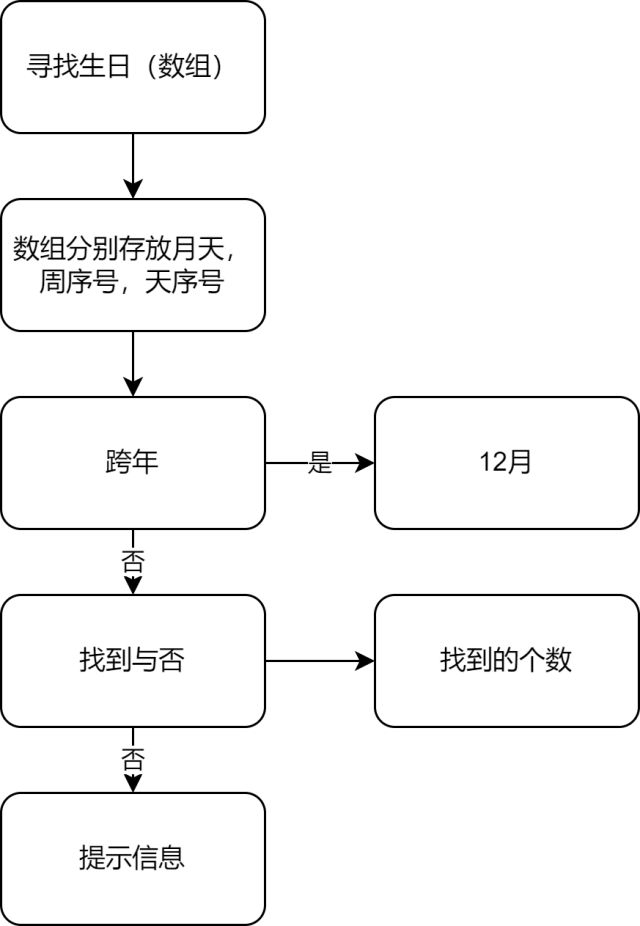
数组中13个元素并使第一个元素为0可以使某些代码更加简化，但也容易出现逻辑错误

# 实验八

## 实验任务



## 实验步骤



## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 正常值 |  |  |
| 2 | 特殊值 |  |  |
| 3 | 跨年 |  |  |

## 实验结论

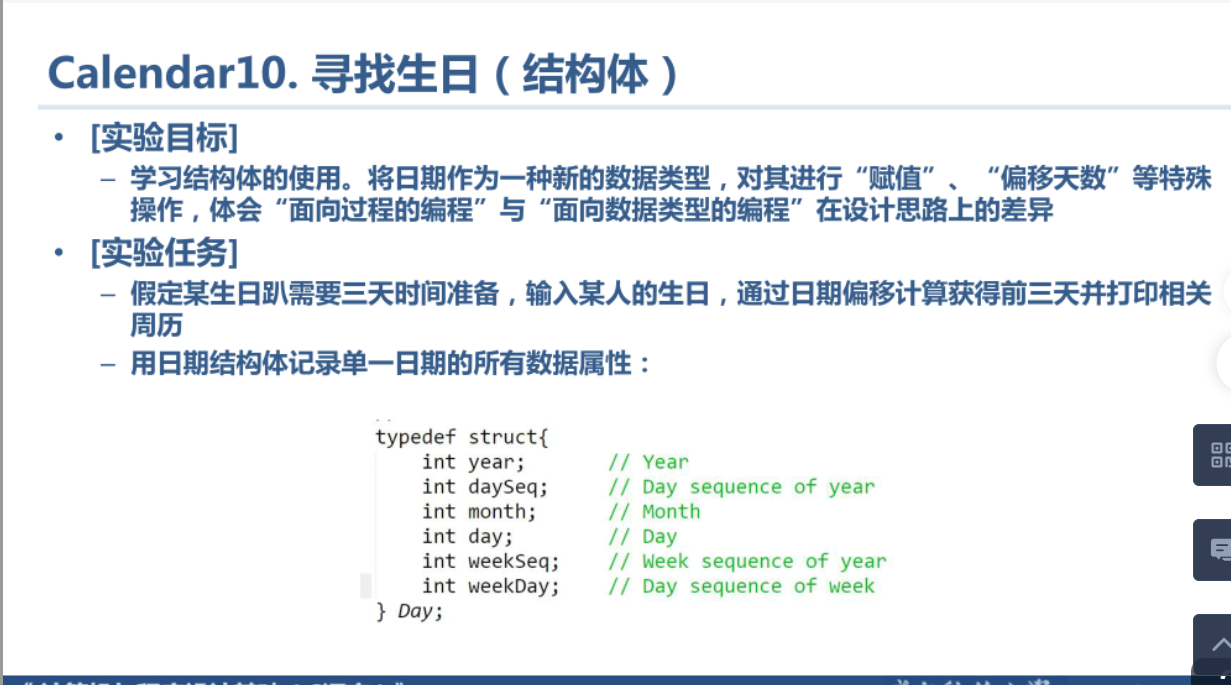
代码达到功能目标

## 实验总结

无

# 实验九

## 实验任务



## 实验步骤

## 未命名绘图

## 代码测试

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 |
| 1 | 对应日期是否正确 |  |  |
| 2 |  |  |  |
| 3 | 是否正确报错 |  |  |

## 实验结论

代码达到功能目标

## 实验总结

需要注意闰年等多种情况

# 本课程学习总结

学习 C 语言的过程宛如一场充满艰辛与惊喜的漫长征途。而在这其中我学到了很多，其中代码的规范性给我留下了最深刻的印象。

在 C 语言的学习与实践中，我深刻领悟到代码规范绝非仅仅是一种可有可无的形式，而是关乎程序生命的核心要素，其重要性怎么强调都不为过。一开始我也喜欢随便命名变量，并且不写注释，过了几天之后就忘记了原本自己的想法。经过本学期刘老师的精心教导，我明白了合理的缩进与空行恰似章节段落间的巧妙分隔，使得代码的逻辑架构一目了然，阅读者能够轻松跟上程序的思维脉络，快速理解其意图。规范的代码在维护升级时还展现出无与伦比的优势。遵循规范的代码即使过了很久，开发人员也能迅速定位到需要修改的部分，高效地进行调整与完善，避免因牵一发而动全身引发的连锁错误，确保程序的稳定性与可靠性，持续为用户提供优质的服务。

# 附录

1. 实验一

（1）main.c

#include <stdio.h>

#include <stdlib.h>

int main()

{

int month;

int day;

int sum;

int a;

printf("Week Day Calculator of Year &d,please input month(1-12) and day(1-31):\n");

scanf("%d%d",&month,&day);

if(month==1){

sum = day;

a = sum%7;

}

else if(month==2){

sum = day+31;

a = sum%7;

}

else if(month==3){

sum = day+60;

a = sum%7;

}

else if(month==4){

sum = day+91;

a = sum%7;

}

else if(month==5){

sum = day+121;

a = sum%7;

}

else if(month==6){

sum = day+152;

a = sum%7;

}

else if(month==7){

sum = day+182;

a = sum%7;

}

else if(month==8){

sum = day+213;

a = sum%7;

}

else if(month==9){

sum = day+244;

a = sum%7;

}

else if(month==10){

sum = day+274;

a=sum%7;

}

else if(month==11){

sum = day+305;

a = sum%7;

}

else{

sum = day+335;

a = sum%7;

}

if(a==0){

a = 7;

}

printf("2024-%d-%d is NO.%d day in that week.\n\n",month,day,a);

return 0;

}

1. 实验二

（1）main.c

#include <stdio.h>

#include <stdlib.h>

int main()

{

int month,day;

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

scanf("%d",&month);

while(month < 1||month > 12)

{

printf("Invalid input.\n");

return 0;

}

int t = 0;

int i = 1;

int num = 1;

int month1;

month1 = month;

switch(month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

day = 31;break;

case 2:

day = 29;break;

default:

day = 30;break;

}

printf("2024-%d has total %d days.\n",month1,day);

int a[12] = {31,29,31,30,31,30,31,31,30,31,30,31};

int d = 0;

int m = 0;

while(month > 1)

{

d = d + a[m];

month--;

m++;

}

d = d % 7;

int temp;

temp = d + 1;

printf("2024-%d-1 is No.%d day in that week.\n",month1,temp);

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

while(i <= 7){

while(d>0)

{

printf(" ");

d--;

i++;

day++;

t++;

}

printf("%10d ",num);

i++;

num++;

t++;

if(t % 7 == 0){

printf("\n");

}

}

while(i <= 14){

printf("%10d ",num);

i++;

num++;

t++;

if(t % 7 == 0){

printf("\n");

}

}

while(i <= 21){

printf("%10d ",num);

i++;

num++;

t++;

if(t % 7 == 0){

printf("\n");

}

}

while(i <= 28){

printf("%10d ",num);

i++;

num++;

t++;

if(t % 7 == 0){

printf("\n");

}

}

while(i <= day){

printf("%10d ",num);

i++;

num++;

t++;

if(t % 7 == 0){

printf("\n");

}

}

return 0;

}

1. 实验三

（1）main.c

#include <stdio.h>

#include <stdlib.h>

int main()

{

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

int week;

scanf("%d",&week);

int d[12]={31,29,31,30,31,30,31,31,30,31,30,31};

if(week >= 1&&week <= 53)

{

int day;

day = week \* 7 - 6;

if(day >= 360)

{

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

}

else

{

printf("No.%d week in 2024 starts from No.%d to No.%d day\n\n",week,day,day + 6);

}

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

int temp,i;

temp = 0;

for(i = 1;i <= 7;i++)

{

while(day>d[temp])

{

day -= d[temp];

temp++;

}

printf("%5s%02d.%02d","",temp + 1,day);

if(temp == 11&&day == 31)

{

break;

}

day += 1;

}

}

else

{

printf("Invalid input!");

}

return 0;

}

1. 实验四

（1）main.c

#include <stdio.h>

#include <stdlib.h>

#define MONTH\_NUM 12

int getMonthLength( int month );

int getDaySeq( int month, int day );

int getMonth( int daySeqOfYear );

int getDay( int daySeqOfYear );

int getDaySeqOfWeek( int daySeqOfYear );

int getNextMonday( int daySeqOfYear );

int getThisSunday( int daySeqOfYear );

void printOneDay( int daySeqOfYear );

int main()

{

int i;

printf("Spring Semester calender of year 2025,please input reference day in Feb(1-28): \n");

scanf("%d",&i);

if(i >= 29||i <=0)

{

printf("Invalid input.\n");

return 0;

}

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

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

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 getMonthLength(int month)

{

int temp;

switch (month)

{

case 1: temp = 31; break;

case 2: temp = 29; break;

case 3: temp = 31; break;

case 4: temp = 30; break;

case 5: temp = 31; break;

case 6: temp = 30; break;

case 7: temp = 31; break;

case 8: temp = 31; break;

case 9: temp = 30; break;

case 10:temp = 31; break;

case 11:temp = 30; break;

case 12:temp = 31; break;

}

return temp;

}

int getDaySeqOfWeek( int daySeqOfYear )

{

daySeqOfYear += 2 ;

daySeqOfYear = daySeqOfYear % 7 ;

return daySeqOfYear ;

}

int getMonth( int dayseq )

{

int temp = 1 ;

while( dayseq > getMonthLength( temp ) )

{

dayseq -= getMonthLength( temp ) ;

temp ++ ;

}

return temp ;

}

int getDaySeq(int month,int day)

{

int temp;

for(temp = 1;temp < month;temp ++)

{

day += getMonthLength(temp);

}

return day;

}

int getNextMonday(int day)

{

if((day + 2) % 7!= 1)

{

if((day + 2) % 7 == 0){

day++;

}

else{

day += (7 - (day + 2) % 7) + 1;

}

}

return day;

}

int getDay( int daySeqOfYear )

{

int i ;

int ret = 0 ;

int monthlength[ MONTH\_NUM + 1 ] = {0} ;

for( i = 1; i <= MONTH\_NUM; i ++)

{

monthlength[ i ] = monthlength[ i - 1 ] + getMonthLength( i );

}

for( i = 0; i <= MONTH\_NUM - 1; i ++)

{

if( daySeqOfYear > monthlength[ i ] && daySeqOfYear <= monthlength[ i + 1 ] )

{

ret = daySeqOfYear - monthlength[ i ] ;

break ;

}

}

return ret ;

}

int getThisSunday(int day)

{

if((day + 2) % 7!= 0){

day += 7 -(day + 2) % 7;

}

return day;

}

void printOneDay(int daySeqOfYear)

{

int day = getDay(daySeqOfYear);

int month = getMonth(daySeqOfYear);

int week = getDaySeqOfWeek(daySeqOfYear);

if(day == 1)

{

if(week == 6 || week == 0)

{

printf("%5s%02d.%01d!"," ",month,day);

}

else

{

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

}

}

else

{

if(week == 6|| week == 0)

{

printf("%9d!",day);

}

else

{

printf("%10d",day);

}

}

}

1. 实验五

（1）main.c

#include <stdio.h>

#include <stdlib.h>

#define MONTH\_NUM 12

int isleapYear(int year);

int getDaySeqOnJan1( int year );

int getMonthLength(int year, int month);

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

int getNextMonday(int year, int day);

int getThisSunday(int year, int day);

int getDay(int year, int daySeqOfYear);

int getDaySeqOfWeek(int year, int daySeqOfYear);

void printOneDay(int year, int daySeqOfYear);

int getMonth(int year, int daySeqOfYear);

int isleapYear(int year) {

int leap;

if (year % 4 == 0) {

if (year % 100 == 0) {

if (year % 400 == 0) {

leap = 1;

} else {

leap = 0;

}

} else {

leap = 1;

}

} else {

leap = 0;

}

return leap;

}

int getDay( int year, int daySeqOfYear )

{

int i ;

int ret = 0 ;

int monthlength[ MONTH\_NUM + 1 ] = {0} ;

for( i = 1; i <= MONTH\_NUM; i ++)

{

monthlength[ i ] = monthlength[ i - 1 ] + getMonthLength( year , i );

}

for( i = 0; i <= MONTH\_NUM - 1; i ++)

{

if( daySeqOfYear > monthlength[ i ] && daySeqOfYear <= monthlength[ i + 1 ] )

{

ret = daySeqOfYear - monthlength[ i ] ;

break ;

}

}

return ret ;

}

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

{

switch( month )

{

case 12:

day += 30 ;

case 11:

day += 31 ;

case 10:

day += 30 ;

case 9:

day += 31 ;

case 8:

day += 31 ;

case 7:

day += 30 ;

case 6:

day += 31 ;

case 5:

day += 30 ;

case 4:

day += 31 ;

case 3:

day += 28 + isleapYear( year ) ;

case 2:

day += 31 ;

break ;

}

return day ;

}

int getDaySeqOfWeek( int year, int daySeqOfYear )

{

daySeqOfYear += getDaySeqOnJan1( year ) - 1 ;

daySeqOfYear = daySeqOfYear % 7 ;

return daySeqOfYear ;

}

int getDaySeqOnJan1( int year )

{

int result ;

result = ( year - 1 +( year - 1) / 4 - ( year - 1)/ 100 +( year - 1)/ 400) % 7 + 1 ;

return result ;

}

int getMonth( int year, int daySeqOfYear )

{

int temp = 1 ;

while( daySeqOfYear > getMonthLength( year, temp ) )

{

daySeqOfYear -= getMonthLength( year, temp ) ;

temp ++ ;

}

return temp ;

}

int getMonthLength( int year , int month )

{

int temp = 0 ;

switch ( month )

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

temp = 31 ;

break ;

case 4:

case 6:

case 9:

case 11:

temp = 30 ;

break ;

case 2:

temp = 28 + isleapYear( year ) ;

break ;

}

return temp ;

}

int getNextMonday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day ++ ;

}

else

{

day += (7 -( day + getDaySeqOnJan1( year ) - 1 ) % 7) + 1 ;

}

}

return day ;

}

int getThisMonday ( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day -= 6 ;

}

else

{

day -= ( day + getDaySeqOnJan1( year ) - 1 ) % 7 - 1 ;

}

}

return day ;

}

int getThisSunday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 0)

{

day += 7 -(day + getDaySeqOnJan1( year ) - 1 ) % 7 ;

}

return day ;

}

void printOneDay( int year, int daySeqOfYear)

{

int day = getDay( year, daySeqOfYear);

int month = getMonth( year, daySeqOfYear);

int week = getDaySeqOfWeek( year, daySeqOfYear);

if(day == 1)

{

if( week == 6|| week == 0)

{

printf("%4s%02d.%02d!"," ", month, day);

}

else

{

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

}

}

else

{

if( week == 6|| week == 0)

{

printf("%9d!", day);

}

else

{

printf("%10d", day);

}

}

}

int main()

{

printf("Fall Semester Calendar of Year \n");

int currentYear ;

scanf("%d", &currentYear);

int currentYearLength = 365 + isleapYear(currentYear);

int sStartSeqOfYear = getNextMonday(currentYear, getDaySeq(currentYear, 9, 4));

int sEndSeqOfYear = getThisSunday(currentYear + 1, getDaySeq(currentYear + 1, 1, 20)) + currentYearLength;

int daySeqOfYear, daySeqOfWeek, weekSeqOfSemester;

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;

}

1. 实验六

（1）main.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

int main()

{

int year ;

printf("School Calendar,please input the year(%d-%d):\n", YEAR\_MIN, YEAR\_MAX) ;

scanf("%d", &year) ;

if( year < YEAR\_MIN || year > YEAR\_MAX )

{

printf("\nInvalid input!\n") ;

return 0 ;

}

printf("%13s%s%d%s%d\n"," ","First Semester (Fall) calendar of Year ", year, " - ", year + 1 ) ;

printf("\n #W:%10s%10s%10s%10s%10s%10s%10s\n"

,"Mon.","Tues.","Wend.","Thur.","Fri.","Sat.","Sun.") ;

int currentyear = 365 + isLeapYear( year ) ;

int sStartSeqOfYear = getThisMonday( year, getDaySeq( year, 9, 4)) ;

int sEndSeqOfYear = currentyear + getThisSunday( year + 1, getDaySeq( year + 1, 1, 20)) ;

int daySeqOfYear, Weeknum ;

int weekSeqOfsemester = ( sEndSeqOfYear - sStartSeqOfYear + 1) / 7 ;

for( Weeknum = 1 , daySeqOfYear = sStartSeqOfYear ;

Weeknum <= weekSeqOfsemester ;

Weeknum ++ , daySeqOfYear += 7 )

{

printoneWeek( year, Weeknum, daySeqOfYear );

}

year ++;

printf("\n%13s%s%d%s%d\n"," ","Second Semester (Spring) calendar of Year ", year - 1, " - ", year ) ;

printf("\n #W:%10s%10s%10s%10s%10s%10s%10s\n"

,"Mon.","Tues.","Wend.","Thur.","Fri.","Sat.","Sun.") ;

sStartSeqOfYear = getThisMonday( year, getDaySeq( year, 2, 15)) ;

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

weekSeqOfsemester = ( sEndSeqOfYear - sStartSeqOfYear + 1) / 7 ;

for( Weeknum = 1 , daySeqOfYear = sStartSeqOfYear ;

Weeknum <= weekSeqOfsemester ;

Weeknum ++ , daySeqOfYear += 7 )

{

printoneWeek( year, Weeknum, daySeqOfYear );

}

return 0 ;

}

（2）date.h

#ifndef DATE\_H\_INCLUDED

#define DATE\_H\_INCLUDED

#define YEAR\_MIN 2018

#define YEAR\_MAX 2024

#define DATE\_INFO\_BRIEF 0

#define DATE\_INFO\_FULL 1

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 day );

int getThisSunday( int year, int daySeqOfYear );

#endif // DATE\_H\_INCLUDED

（3）funs.h

#ifndef FUNS\_H\_INCLUDED

#define FUNS\_H\_INCLUDED

void printOneDay( int year, int daySeqOfYear , int formatType );

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

#endif // FUNS\_H\_INCLUDED

1. date.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

#define MONTH\_NUM 12

int getDay( int year, int daySeqOfYear )

{

int i ;

int ret = 0 ;

int monthlength[ MONTH\_NUM + 1 ] = {0} ;

for( i = 1; i <= MONTH\_NUM; i ++)

{

monthlength[ i ] = monthlength[ i - 1 ] + getMonthLength( year , i );

}

for( i = 0; i <= MONTH\_NUM - 1; i ++)

{

if( daySeqOfYear > monthlength[ i ] && daySeqOfYear <= monthlength[ i + 1 ] )

{

ret = daySeqOfYear - monthlength[ i ] ;

break ;

}

}

return ret ;

}

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

{

switch( month )

{

case 12:

day += 30 ;

case 11:

day += 31 ;

case 10:

day += 30 ;

case 9:

day += 31 ;

case 8:

day += 31 ;

case 7:

day += 30 ;

case 6:

day += 31 ;

case 5:

day += 30 ;

case 4:

day += 31 ;

case 3:

day += 28 + isLeapYear( year ) ;

case 2:

day += 31 ;

break ;

}

return day ;

}

int getDaySeqOfWeek( int year, int daySeqOfYear )

{

daySeqOfYear += getDaySeqOnJan1( year ) - 1 ;

daySeqOfYear = daySeqOfYear % 7 ;

return daySeqOfYear ;

}

int getDaySeqOnJan1( int year )

{

int result ;

result = ( year - 1 +( year - 1) / 4 - ( year - 1)/ 100 +( year - 1)/ 400) % 7 + 1 ;

return result ;

}

int getMonth( int year, int dayseq )

{

int temp = 1 ;

while( dayseq > getMonthLength( year, temp ) )

{

dayseq -= getMonthLength( year, temp ) ;

temp ++ ;

}

return temp ;

}

int getMonthLength( int year , int month )

{

int monthnum = 0 ;

switch ( month )

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

monthnum = 31 ;

break ;

case 4:

case 6:

case 9:

case 11:

monthnum = 30 ;

break ;

case 2:

monthnum = 28 + isLeapYear( year ) ;

break ;

}

return monthnum ;

}

int getNextMonday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day ++ ;

}

else

{

day += (7 -( day + getDaySeqOnJan1( year ) - 1 ) % 7) + 1 ;

}

}

return day ;

}

int getThisMonday ( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day -= 6 ;

}

else

{

day -= ( day + getDaySeqOnJan1( year ) - 1 ) % 7 - 1 ;

}

}

/\* day = getNextMonday ( year, day) - 7 ; \*/

return day ;

}

int getThisSunday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 0)

{

day += 7 -(day + getDaySeqOnJan1( year ) - 1 ) % 7 ;

}

return day ;

}

int isLeapYear( int year )

{

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

{

return 1 ;

}

return 0 ;

}

1. funs.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

void printOneDay( int year, int daySeqOfYear , int formatType )

{

int day = getDay( year, daySeqOfYear);

int month = getMonth( year, daySeqOfYear);

int week = getDaySeqOfWeek( year, daySeqOfYear);

if( formatType == 1 )

{

printf("%2s%02d.%02d.%02d"," ", year % 100 , month, day);

}

else if( day == 1)

{

if( month == 1)

{

if( week == 6|| week == 0)

{

printf("%1s%02d.%02d.%02d!"," ", year % 100 , month, day);

}

else

{

printf("%2s%02d.%02d.%02d"," ", year % 100 , month, day);

}

}

else if( week == 6|| week == 0)

{

printf("%4s%02d.%02d!"," ", month, day);

}

else

{

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

}

}

else

{

if( week == 6|| week == 0)

{

printf("%9d!", day);

}

else

{

printf("%10d", day);

}

}

}

void printoneWeek(int year, int weekSeqOfYear, int weekSeqShow )

{

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

int day = 1 ;

if( weekSeqOfYear == 1)

{

printOneDay( year, weekSeqShow , DATE\_INFO\_FULL ) ;

day ++ ;

weekSeqShow ++ ;

}

int currentyear = 365 + isLeapYear( year ) ;

for( ; day <= 7 ; day ++ , weekSeqShow ++ )

{

if( weekSeqShow <= currentyear )

{

printOneDay( year, weekSeqShow , DATE\_INFO\_BRIEF ) ;

}

else

{

printOneDay( year + 1, weekSeqShow - currentyear , DATE\_INFO\_BRIEF ) ;

}

}

printf("\n") ;

}

1. 实验七

（1）main.c

（2）date.h

#ifndef DATE\_H\_INCLUDED

#define DATE\_H\_INCLUDED

#define YEAR\_MIN 2000

#define YEAR\_MAX 2030

#define YEAR\_NUM 5

#define DATE\_INFO\_BRIEF 0

#define DATE\_INFO\_FULL 1

int isLeapYear( int year );

int getDaySeqOnJan1( int year );

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 day );

int getThisSunday( int year, int daySeqOfYear );

void setYearArray( int Years[], int yearnum, int year );

#endif // DATE\_H\_INCLUDED

（3）funs.h

#ifndef FUNS\_H\_INCLUDED

#define FUNS\_H\_INCLUDED

void printOneDay( int year, int daySeqOfYear , int formatType );

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

#endif // FUNS\_H\_INCLUDED

1. date.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

#define MONTH\_NUM 12

int getDay( int year, int daySeqOfYear )

{

extern int Month\_LEAP\_YEAR[12];

extern int Month\_NORMAL\_YEAR[12];

int i = 0 ;

int ret = daySeqOfYear ;

if( isLeapYear( year ) == 1 )

{

while( ret > Month\_LEAP\_YEAR[ i ] )

{

ret -= Month\_LEAP\_YEAR[ i ] ;

i ++ ;

}

}

else

{

while( ret > Month\_NORMAL\_YEAR[ i ] )

{

ret -= Month\_NORMAL\_YEAR[ i ] ;

i ++ ;

}

}

return ret ;

}

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

{

switch( month )

{

case 12:

day += 30 ;

case 11:

day += 31 ;

case 10:

day += 30 ;

case 9:

day += 31 ;

case 8:

day += 31 ;

case 7:

day += 30 ;

case 6:

day += 31 ;

case 5:

day += 30 ;

case 4:

day += 31 ;

case 3:

day += 28 + isLeapYear( year ) ;

case 2:

day += 31 ;

break ;

}

return day ;

}

int getDaySeqOfWeek( int year, int daySeqOfYear )

{

daySeqOfYear += getDaySeqOnJan1( year ) - 1 ;

daySeqOfYear = daySeqOfYear % 7 ;

return daySeqOfYear ;

}

int getDaySeqOnJan1( int year )

{

int result ;

result = ( year - 1 +( year - 1) / 4 - ( year - 1)/ 100 +

( year - 1)/ 400) % 7 + 1 ;

return result ;

}

int getMonth( int year, int dayseq )

{

extern int Month\_LEAP\_YEAR[12];

extern int Month\_NORMAL\_YEAR[12];

int temp = 1 ;

if( isLeapYear( year ) == 1 )

{

while( dayseq > Month\_LEAP\_YEAR[ temp - 1 ] )

{

dayseq -= Month\_LEAP\_YEAR[ temp - 1 ] ;

temp ++ ;

}

}

else

{

while( dayseq > Month\_NORMAL\_YEAR[ temp - 1 ] )

{

dayseq -= Month\_NORMAL\_YEAR[ temp - 1 ] ;

temp ++ ;

}

}

return temp ;

}

int getNextMonday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day ++ ;

}

else

{

day += (7 -( day + getDaySeqOnJan1( year ) - 1 ) % 7) + 1 ;

}

}

return day ;

}

int getThisMonday ( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day -= 6 ;

}

else

{

day -= ( day + getDaySeqOnJan1( year ) - 1 ) % 7 - 1 ;

}

}

return day ;

}

int getThisSunday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 0)

{

day += 7 -(day + getDaySeqOnJan1( year ) - 1 ) % 7 ;

}

return day ;

}

int isLeapYear( int year )

{

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

{

return 1 ;

}

return 0 ;

}

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

{

int week ;

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

int currentyear = 365 + isLeapYear( year ) ;

if( daySeqOfYear > currentyear )

{

if( getDaySeqOfWeek( year + 1, daySeqOfYear - currentyear) == 0)

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) ) / 7 ;

}

else

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) +

( 7 - getDaySeqOfWeek( year + 1, daySeqOfYear - currentyear))) / 7 ;

}

}

else

{

if( getDaySeqOfWeek( year , daySeqOfYear ) == 0)

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) ) / 7 ;

}

else

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) +

( 7 - getDaySeqOfWeek( year , daySeqOfYear)))/ 7 ;

}

}

return week;

}

void setYearArray( int Years[], int yearnum, int year )

{

int num1 ;

for( num1 = 0; num1 < yearnum ; num1 ++ , year ++)

{

Years[ num1 ] = year;

}

}

1. funs.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

void printOneDay( int year, int daySeqOfYear , int formatType )

{

int day = getDay( year, daySeqOfYear);

int month = getMonth( year, daySeqOfYear);

int week = getDaySeqOfWeek( year, daySeqOfYear);

if( formatType == 1 )

{

printf("%2s%02d.%02d.%02d"," ", year % 100 , month, day);

}

else if( day == 1)

{

if( month == 1)

{

if( week == 6|| week == 0)

{

printf("%1s%02d.%02d.%02d!"," ", year % 100 , month, day);

}

else

{

printf("%2s%02d.%02d.%02d"," ", year % 100 , month, day);

}

}

else if( week == 6|| week == 0)

{

printf("%4s%02d.%02d!"," ", month, day);

}

else

{

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

}

}

else

{

if( week == 6|| week == 0)

{

printf("%9d!", day);

}

else

{

printf("%10d", day);

}

}

}

void printoneWeek(int year, int weekSeqOfYear, int weekSeqShow )

{

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

int sStartSeqOfYear = 7 \* ( weekSeqOfYear - 1 ) - getDaySeqOnJan1( year ) + 2 ;

int sEndSeqOfYear = sStartSeqOfYear + 6 ;

int currentyear = 365 + isLeapYear( year ) ;

int daySeqOfYear = sStartSeqOfYear ;

if( weekSeqShow == 1)

{

printOneDay( year, daySeqOfYear , DATE\_INFO\_FULL ) ;

daySeqOfYear ++ ;

}

for( ; daySeqOfYear <= sEndSeqOfYear ; daySeqOfYear ++ )

{

if( daySeqOfYear <= currentyear )

{

printOneDay( year, daySeqOfYear , DATE\_INFO\_BRIEF ) ;

}

else

{

printOneDay( year + 1, daySeqOfYear - currentyear , DATE\_INFO\_BRIEF ) ;

}

}

printf("\n") ;

}

1. 实验八
2. main.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

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 main()

{

//Declare variables

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

//Declare a three-dimension array to record date information.int Days[YEAR\_NUM][366][4]={0]};

// D1splay the program information

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 > 31)

{

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

return 1;

}

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

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

initialDays ( Years, Days, YEAR\_NUM);

int totalNum = 0;

int weekendNum = 0;

//Requirement: finish the rest functions

// 1. the print related sentences:

// printf("\n%s%sNd\n","","Birthday in Year ", year );

// printf("#M:%10s%10s%10s%10s%10s%105%10s\n",

// "Mon.","Tues.","Wed.","Thur.","Fri.","Sat.","Sun.");

// printf(" Not found.\n");

// 2. the print related function calling:// printOneWeek (year, weekSeq, month);/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//<< Your source code starts here >>//

//

int i,j;

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

{

int found = 0;

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

for (j = 0; j < (isLeapYear(Years[i]) ? 366 : 365); j ++)

{

if (Days[i][j][0] == inputMonth && Days[i][j][1] == inputDay)

{

totalNum ++;

found = 1;

printf("#M:%10s%10s%10s%10s%10s%10s%10s\n",

"Mon.", "Tues.", "Wed.", "Thur.", "Fri.", "Sat.", "Sun.");

printoneWeek(Years[i], getWeekSeqOfYear(Years[i], inputMonth, inputDay), inputMonth);

if (Days[i][j][3] == 6 || Days[i][j][3] == 0)

{

weekendNum++;

}

break;

}

}

if (!found)

{

printf(" Not found.\n");

}

}

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

return 0;

}

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

{

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

int i, j;

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

}

}

}

1. date.h

#ifndef DATE\_H\_INCLUDED

#define DATE\_H\_INCLUDED

#define YEAR\_MIN 2019

#define YEAR\_MAX 2026

#define YEAR\_NUM 8

#define DATE\_INFO\_BRIEF 0

#define DATE\_INFO\_FULL 1

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 day );

int getThisSunday( int year, int daySeqOfYear );

void setYearArray( int Years[], int yearnum, int year );

#endif // DATE\_H\_INCLUDED

1. funs.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

void printOneDay( int year, int daySeqOfYear , int formatType )

{

int day = getDay( year, daySeqOfYear);

int month = getMonth( year, daySeqOfYear);

int week = getDaySeqOfWeek( year, daySeqOfYear);

if(day <= 0)

{

if( week == 6|| week == 0)

{

printf("%4s%02d.%02d!"," ", 12, day + 31);

}

else

{

printf("%5s%02d.%02d"," ", 12, day + 31);

}

}

else if( formatType == 1 )

{

printf("%2s%02d.%02d.%02d"," ", year % 100 , month, day);

}

else if( day == 1)

{

if( month == 1)

{

if( week == 6|| week == 0)

{

printf("%1s%02d.%02d.%02d!"," ", year % 100 , month, day);

}

else

{

printf("%2s%02d.%02d.%02d"," ", year % 100 , month, day);

}

}

else if( week == 6|| week == 0)

{

printf("%4s%02d.%02d!"," ", month, day);

}

else

{

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

}

}

else

{

if( week == 6|| week == 0)

{

printf("%9d!", day);

}

else

{

printf("%10d", day);

}

}

}

void printoneWeek(int year, int weekSeqOfYear, int weekSeqShow )

{

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

int sStartSeqOfYear = 7 \* ( weekSeqOfYear - 1 ) - getDaySeqOnJan1( year ) + 2 ;

int sEndSeqOfYear = sStartSeqOfYear + 6 ;

int currentyear = 365 + isLeapYear( year ) ;

int daySeqOfYear = sStartSeqOfYear ;

if( weekSeqShow == 1)

{

printOneDay( year, daySeqOfYear , DATE\_INFO\_FULL ) ;

daySeqOfYear ++ ;

}

for( ; daySeqOfYear <= sEndSeqOfYear ; daySeqOfYear ++ )

{

if( daySeqOfYear <= currentyear )

{

printOneDay( year, daySeqOfYear , DATE\_INFO\_BRIEF ) ;

}

else

{

printOneDay( year + 1, daySeqOfYear - currentyear , DATE\_INFO\_BRIEF ) ;

}

}

printf("\n") ;

}

1. date.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

#define MONTH\_NUM 12

int getDay( int year, int daySeqOfYear )

{

extern int Month\_LEAP\_YEAR[12];

extern int Month\_NORMAL\_YEAR[12];

int i = 0 ;

int ret = daySeqOfYear ;

if( isLeapYear( year ) == 1 )

{

while( ret > Month\_LEAP\_YEAR[ i ] )

{

ret -= Month\_LEAP\_YEAR[ i ] ;

i ++ ;

}

}

else

{

while( ret > Month\_NORMAL\_YEAR[ i ] )

{

ret -= Month\_NORMAL\_YEAR[ i ] ;

i ++ ;

}

}

return ret ;

}

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

{

switch( month )

{

case 12:

day += 30 ;

case 11:

day += 31 ;

case 10:

day += 30 ;

case 9:

day += 31 ;

case 8:

day += 31 ;

case 7:

day += 30 ;

case 6:

day += 31 ;

case 5:

day += 30 ;

case 4:

day += 31 ;

case 3:

day += 28 + isLeapYear( year ) ;

case 2:

day += 31 ;

break ;

}

return day ;

}

int getDaySeqOfWeek( int year, int daySeqOfYear )

{

daySeqOfYear += getDaySeqOnJan1( year ) - 1 ;

daySeqOfYear = daySeqOfYear % 7 ;

return daySeqOfYear ;

}

int getDaySeqOnJan1( int year )

{

int result ;

result = ( year - 1 +( year - 1) / 4 - ( year - 1)/ 100 +( year - 1)/ 400) % 7 + 1 ;

return result ;

}

int getMonth( int year, int dayseq )

{

extern int Month\_LEAP\_YEAR[12];

extern int Month\_NORMAL\_YEAR[12];

int temp = 1 ;

if( isLeapYear( year ) == 1 )

{

while( dayseq > Month\_LEAP\_YEAR[ temp - 1 ] )

{

dayseq -= Month\_LEAP\_YEAR[ temp - 1 ] ;

temp ++ ;

}

}

else

{

while( dayseq > Month\_NORMAL\_YEAR[ temp - 1 ] )

{

dayseq -= Month\_NORMAL\_YEAR[ temp - 1 ] ;

temp ++ ;

}

}

return temp ;

}

/\*int getMonthLength( int year , int month )

{

int monthnum = 0 ;

switch ( month )

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

monthnum = 31 ;

break ;

case 4:

case 6:

case 9:

case 11:

monthnum = 30 ;

break ;

case 2:

monthnum = 28 + isLeapYear( year ) ;

break ;

}

return monthnum ;

}\*/

int getNextMonday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day ++ ;

}

else

{

day += (7 -( day + getDaySeqOnJan1( year ) - 1 ) % 7) + 1 ;

}

}

return day ;

}

int getThisMonday ( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 1)

{

if(( day + getDaySeqOnJan1( year ) - 1 ) % 7 == 0)

{

day -= 6 ;

}

else

{

day -= ( day + getDaySeqOnJan1( year ) - 1 ) % 7 - 1 ;

}

}

/\* day = getNextMonday ( year, day) - 7 ; \*/

return day ;

}

int getThisSunday( int year, int day )

{

if( ( day + getDaySeqOnJan1( year ) - 1 ) % 7 != 0)

{

day += 7 -(day + getDaySeqOnJan1( year ) - 1 ) % 7 ;

}

return day ;

}

int isLeapYear( int year )

{

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

{

return 1 ;

}

return 0 ;

}

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

{

int week ;

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

int currentyear = 365 + isLeapYear( year ) ;

if( daySeqOfYear > currentyear )

{

if( getDaySeqOfWeek( year + 1, daySeqOfYear - currentyear) == 0)

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) ) / 7 ;

}

else

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) +

( 7 - getDaySeqOfWeek( year + 1, daySeqOfYear - currentyear))) / 7 ;

}

}

else

{

if( getDaySeqOfWeek( year , daySeqOfYear ) == 0)

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) ) / 7 ;

}

else

{

week = ( daySeqOfYear + getDaySeqOnJan1( year ) +

( 7 - getDaySeqOfWeek( year , daySeqOfYear)))/ 7 ;

}

}

return week;

}

void setYearArray( int Years[], int yearnum, int year )

{

int num1 ;

for( num1 = 0; num1 < yearnum ; num1 ++ , year ++)

{

Years[ num1 ] = year;

}

}

1. funs.h

#ifndef FUNS\_H\_INCLUDED

#define FUNS\_H\_INCLUDED

void printOneDay( int year, int daySeqOfYear , int formatType );

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

#endif // FUNS\_H\_INCLUDED

1. 实验九

（1）main.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

const int Years[5] = {2018,2019,2020,2021,2022};

int main()

{

int inputMonth,inputDay;

int i = 0;

Day birthDay = {0};

Day prepareDay = {0};

Day printDay = {0};

printf("Finding Birthday in year(2018-2022),Please input month and day: \n");

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

printf("\n\n");

int j = 0;

if(inputMonth > 12||inputMonth < 1||inputDay > 31||inputDay < 0)

{

printf("Invalid input.");

return 0;

}

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

{

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

prepareDay = getDayBefore (birthDay, printDayRange);

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

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

{

printf(" Not found!\n");

continue;

}

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

printoneWeek(prepareDay);

printf("\n");

}

return 0 ;

}

（2）date.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

#define MONTH\_NUM 12

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 getDay( int year, int daySeqOfYear )

{

int i = 0 ;

int ret = daySeqOfYear ;

if( isLeapYear( year ) == 1 )

{

while( ret > Month\_LEAP\_YEAR[ i ] )

{

ret -= Month\_LEAP\_YEAR[ i ] ;

i ++ ;

}

}

else

{

while( ret > Month\_NORMAL\_YEAR[ i ] )

{

ret -= Month\_NORMAL\_YEAR[ i ] ;

i ++ ;

}

}

return ret ;

}

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

{

switch( month )

{

case 12:

day += 30 ;

case 11:

day += 31 ;

case 10:

day += 30 ;

case 9:

day += 31 ;

case 8:

day += 31 ;

case 7:

day += 30 ;

case 6:

day += 31 ;

case 5:

day += 30 ;

case 4:

day += 31 ;

case 3:

day += 28 + isLeapYear( year ) ;

case 2:

day += 31 ;

break ;

}

return day ;

}

int getDaySeqOfWeek( int year, int daySeqOfYear )

{

daySeqOfYear += getDaySeqOnJan1( year ) - 1 ;

daySeqOfYear = daySeqOfYear % 7 ;

return daySeqOfYear ;

}

int getDaySeqOnJan1( int year )

{

int result ;

result = ( year - 1 +( year - 1) / 4 - ( year - 1)/ 100 +( year - 1)/ 400) % 7 + 1 ;

return result ;

}

int getMonth( int year, int dayseq )

{

int temp = 1 ;

if( isLeapYear( year ) == 1 )

{

while( dayseq > Month\_LEAP\_YEAR[ temp - 1 ] )

{

dayseq -= Month\_LEAP\_YEAR[ temp - 1 ] ;

temp ++ ;

}

}

else

{

while( dayseq > Month\_NORMAL\_YEAR[ temp - 1 ] )

{

dayseq -= Month\_NORMAL\_YEAR[ temp - 1 ] ;

temp ++ ;

}

}

return temp ;

}

（3）date.h

#ifndef DATE\_H\_INCLUDED

#define DATE\_H\_INCLUDED

#define YEAR\_MIN 2000

#define YEAR\_MAX 2030

#define YEAR\_NUM 5

#define DATE\_INFO\_BRIEF 0

#define DATE\_INFO\_FULL 1

#define YEAR\_NUM 5

#define printDayRange 3

typedef struct day{

int year;

int dayseq;

int month;

int day;

int weekseq;

int weekDay;

} Day;

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 day );

int getThisSunday( int year, int daySeqOfYear );

void setYearArray( int Years[], int yearnum, int year );

int getweekDay(int year,int daySeq);

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 );

#endif // DATE\_H\_INCLUDED

（4）funs.c

#include <stdio.h>

#include <stdlib.h>

#include "date.h"

#include "funs.h"

extern int Month\_NORMAL\_YEAR[12];

extern int Month\_LEAP\_YEAR[12];

void printoneWeek(Day prepareday) {

Day remain = prepareday;

int index = 1;

int j,k;

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

// 处理周序号

if (remain.dayseq + 7 - remain.weekDay > isLeapYear(prepareday.year) + 365) {

prepareday.weekseq = 1;

}

printf("[%02d]", prepareday.weekseq);

// 打印空格

for (j = 1; j < remain.weekDay; j++) {

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

}

for (k = 0; k <= printDayRange; k++) {

if (k == 0) {

printf(" %02d.%02d.%02d", prepareday.year - 100 \* (prepareday.year / 100), prepareday.month, prepareday.day);

prepareday.day++;

prepareday.weekseq = getWeekSeqOfYear(prepareday.year, prepareday.month, prepareday.day);

} else {

// 判断是否需要换行

if (k + remain.weekDay > 7 \* index) {

getWeekSeqOfYear(prepareday.year, prepareday.month, getThisSunday(prepareday.year, prepareday.dayseq));

if (prepareday.dayseq + 7 > isLeapYear(prepareday.year) + 365) {

prepareday.weekseq = 1;

}

printf("\n");

printf("[%02d]", prepareday.weekseq);

index++;

}

// 判断是否为生日当天

if (k == printDayRange) {

if (remain.dayseq + k > isLeapYear(prepareday.year) + 365) {

prepareday.month = 1;

prepareday.day = 1;

prepareday.year++;

prepareday.dayseq = 1;

prepareday.weekseq = 0;

} else if (prepareday.month == 2 && isLeapYear(prepareday.year) == 1 && prepareday.day == 29) {

prepareday.month++;

prepareday.day = 1;

prepareday.dayseq++;

} else if (prepareday.day > Month\_NORMAL\_YEAR[prepareday.month - 1]) {

prepareday.day = 1;

prepareday.month++;

}

if (k == printDayRange) {

// 打印生日当天带\*

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

} else {

// 非生日当天正常打印

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

}

prepareday.day++;

prepareday.dayseq++;

prepareday.weekseq = getWeekSeqOfYear(prepareday.year, prepareday.month, prepareday.day);

} else {

if (remain.dayseq + k > isLeapYear(prepareday.year) + 365) {

prepareday.month = 1;

prepareday.day = 1;

prepareday.year++;

prepareday.dayseq = 1;

prepareday.weekseq = 0;

} else if (prepareday.month == 2 && isLeapYear(prepareday.year) == 1 && prepareday.day == 29) {

prepareday.month++;

prepareday.day = 1;

prepareday.dayseq++;

} else if (prepareday.day > Month\_NORMAL\_YEAR[prepareday.month - 1]) {

prepareday.day = 1;

prepareday.month++;

}

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

prepareday.day++;

prepareday.dayseq++;

prepareday.weekseq = getWeekSeqOfYear(prepareday.year, prepareday.month, prepareday.day);

}

}

}

printf("\n");

}

（5）funs.h

#ifndef FUNS\_H\_INCLUDED

#define FUNS\_H\_INCLUDED

void printoneWeek( Day day);

#endif // FUNS\_H\_INCLUDED