例5-8：设计万年历（非函数实现版本）

```c
#include <stdio.h>
int main()
{
    int year, month, day, weekday, monthDays, i, n, leap=0;
    printf("\n请输入某年年份: ");
    scanf("%d",&year);

    n = year - 1900;//求元旦是星期几
    n = (n+(n-1)/4+1)%7;
    weekday = n;

    if((year%4==0 && year%100!=0)||year%400==0)//判断是否是闰年
        leap = 1;

    printf("\n\n\t\t%d\n\n", year);
    for(month=1; month<=12; month=month+1) //打印12个月的月历
    {
        printf("\n%d月份\n",month);
        printf("-----------------------------------------------\n");
        printf("星期日  星期一  星期二  星期三  星期四  星期五  星期六\n");
        printf("-----------------------------------------------\n");
        for(i=0; i<weekday; i=i+1) //找当月1日的打印位置
            printf("        ");

        if(month == 4 || month == 6 || month == 9 || month == 11)
            monthDays = 30;
        else if(month == 2)
        {
            if(leap == 1)
                monthDays = 29;
            else
                monthDays = 28;
        }
        else
            monthDays = 31;

        for(day=1; day<=monthDays; day++) //打印当月日期
        {
            printf("   %2d   ", day);
            weekday++;
            if(weekday==7) //打满一星期应换行
            {
                weekday = 0;
                printf("\n");
            }
        }
        if(weekday)
            printf("\n");//打完一月应换行
    }

    return 0;
}
```

贴图版（防拷贝）

非函数版本

代码略长，可读性、可维护性、可扩展性，都不太好！

运行程序

```
命令提示符
D:\a1ac\example\chap5>c5-8_CalendarMain

请输入某年年份: 2024

                2024

1月份
-----------------------------------------------
星期日  星期一  星期二  星期三  星期四  星期五  星期六
-----------------------------------------------
                1       2       3       4       5       6
    7       8       9      10      11      12      13
   14      15      16      17      18      19      20
   21      22      23      24      25      26      27
   28      29      30      31

10月份
-----------------------------------------------
星期日  星期一  星期二  星期三  星期四  星期五  星期六
-----------------------------------------------
                        1       2       3       4       5
    6       7       8       9      10      11      12
   13      14      15      16      17      18      19
   20      21      22      23      24      25      26
   27      28      29      30      31

11月份
-----------------------------------------------
星期日  星期一  星期二  星期三  星期四  星期五  星期六
-----------------------------------------------
                                        1       2
    3       4       5       6       7       8       9
   10      11      12      13      14      15      16
   17      18      19      20      21      22      23
   24      25      26      27      28      29      30
```

例5-9：设计万年历（函数实现的构思）

```c
#include <stdio.h>
int main()
{
    int year, month, day, weekday, monthDays, i, n, leap=0;
    printf("\n请输入某年年份: ");
    scanf("%d",&year);

    n = year - 1900;//求元旦是星期几
    n = (n+(n-1)/4+1)%7;
    weekday = n;

    if((year%4==0 && year%100!=0)||year%400==0)//判断是否是闰年
        leap = 1;

    printf("\n\n\t\t%d\n\n", year);
    for(month=1; month<=12; month=month+1) //打印12个月的月历
    {
        if(month == 4 || month == 6 || month == 9 || month == 11)
            monthDays = 30;
        else if(month == 2)
        {
            if(leap == 1)
                monthDays = 29;
            else
                monthDays = 28;
        }
        else
            monthDays = 31;

        printf("\n%d月份\n",month);
        printf("-----------------------------------------------------\n");
        printf("星期日  星期一  星期二  星期三  星期四  星期五  星期六\n");
        printf("-----------------------------------------------------\n");
        for(i=0; i<weekday; i=i+1) //找当月1日的打印位置
            printf("        ");

        for(day=1; day<=monthDays; day++) //打印当月日期
        {
            printf("   %2d   ", day);
            weekday++;
            if(weekday==7) //打满一星期应换行
            {
                weekday = 0;
                printf("\n");
            }
        }
        if(weekday)
            printf("\n");//打完一月应换行
    }

    return 0;
}
```

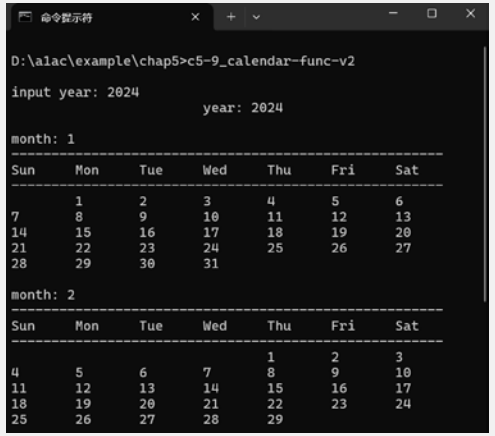
这五个地方用函数来实现会更清晰，更易于维护！

```c
int getWeekDay(int year, int month, int day);   //查询某天是星期几
int isLeap(int year);   //判断是否是闰年
int getDaysOfMonth(int year, int month);        // 获得月天数
void printYearCalender(int year);   // 打印年历
void printMonthCalendar(int month, int days, int weekday);

int main() // main函数清清爽爽
{
    int year, weekday;
    printf("\ninput year: ");
    scanf("%d", &year);

    printYearCalender(year); // 打印年历

    return 0;
}
```

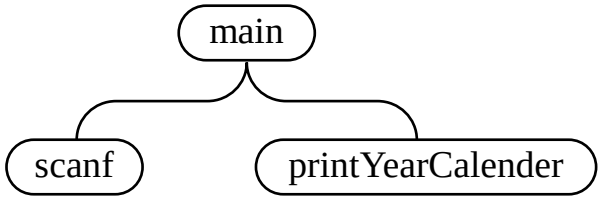
作业：讲完后面几页后，把这五个地方用五个函数来实现！

*例5-9：设计万年历（函数实现）

```c
void printYearCalender(int year);                    // 打印年历
void printMonthCalendar(int month, int days, int weekday);
int getWeekDay(int year, int month, int day);  //查询某天是星期几
int isLeap(int year);   //判断是否是闰年
int getDaysOfMonth(int year, int month);          // 获得月天数

int main() // main函数清清爽爽
{
    int year, weekday;

    printf("\ninput year: ");
    scanf("%d", &year);

    printYearCalender(year); //打印年历

    return 0;
}
```



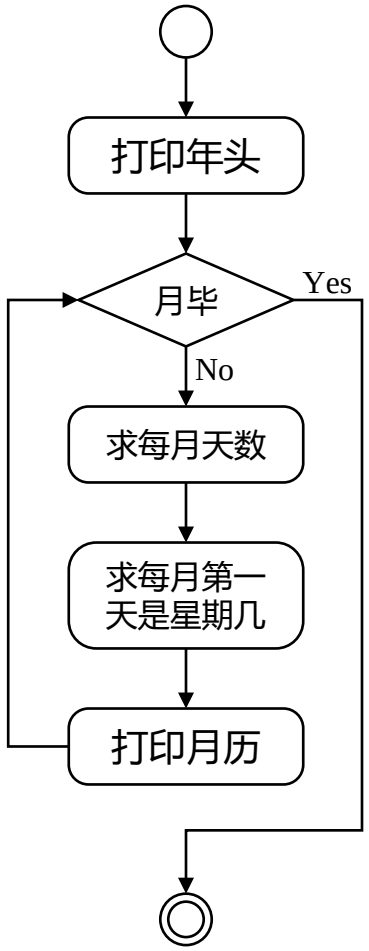
主程序就两行：输入和输出



例5-9：设计万年历（函数实现）

```
命令提示符

D:\a1ac\example\chap5>c5-9_calendar-func-v2

input year: 2024
                        year: 2024

month: 1
------------------------------------------------------
Sun     Mon     Tue     Wed     Thu     Fri     Sat
------------------------------------------------------
        1       2       3       4       5       6
7       8       9       10      11      12      13
14      15      16      17      18      19      20
21      22      23      24      25      26      27
28      29      30      31

month: 2
------------------------------------------------------
Sun     Mon     Tue     Wed     Thu     Fri     Sat
------------------------------------------------------
                                1       2       3
4       5       6       7       8       9       10
11      12      13      14      15      16      17
18      19      20      21      22      23      24
25      26      27      28      29
```

⋮




```
void printYearCalender(int year)
{
    int days, month, weekday;
    printf("\t\t\tyear: %d\n", year);
    for(month = 1; month <= 12; month++)
    {
        days = getDaysOfMonth(year, month);    // 每月多少天?
        weekday = getWeekDay(year, month, 1); // 月初星期几?
        printMonthCalendar(month, days, weekday); //打印月历
    }
}
```

解题思路：

(1) 逐月打印一年的所有天。

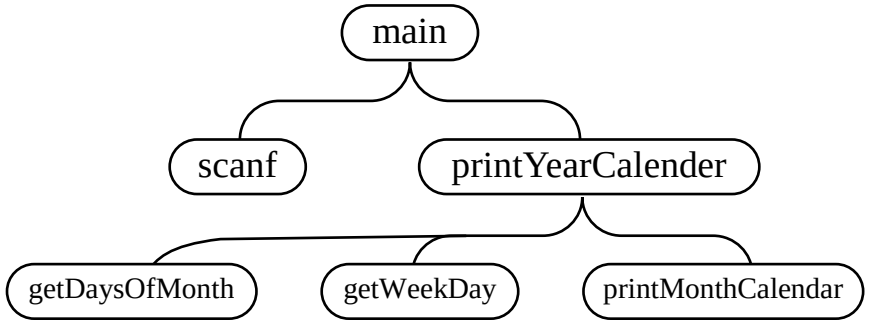
不难，但是很繁琐！
能少写点就太好了！
代码的复用是关键！

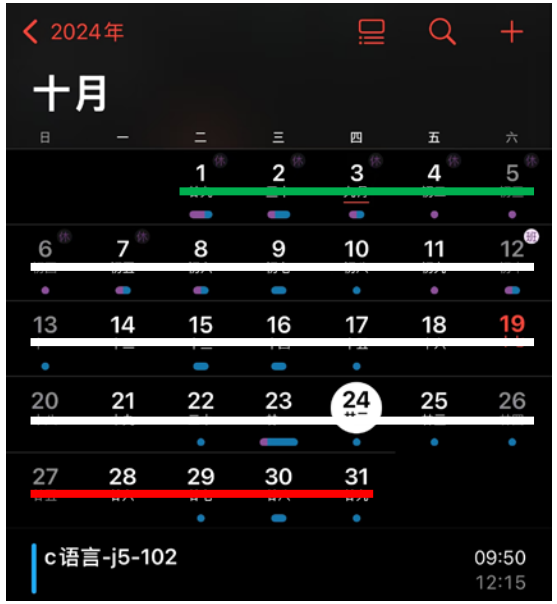
例5-9：设计万年历（函数实现）

```c
void printYearCalender(int year)
{
    int days, month, weekday;
    printf("\t\t\tyear: %d\n", year);
    for(month = 1; month <= 12; month++)
    {
        days = getDaysOfMonth(year, month);       // 返回值：某年某月有多少天
        weekday = getWeekDay(year, month, 1);     // 返回值：每个月的第一天是星期几
        printMonthCalendar(month, days, weekday);// 打印月历，依序打印每个月，从每月的第一天开始
    }
}
void printMonthCalendar(int month, int days, int weekday)
{
    int monthday, i;
    printf("\nmonth: %d\n", month);     // ........................... // 打印月头
    printf("-----------------------------------------------------\n"); // 打印星期表格头
    printf("Sun\tMon\tTue\tWed\tThu\tFri\tSat\n");
    printf("-----------------------------------------------------\n");

    for (i = 1; i <= days + weekday; i++)
        i <= weekday ? putchar('\t') : printf((i % 7) ? "%d\t" : "%d\n", i - weekday);

    putchar('\n');
}
int getDaysOfMonth(int year, int month) //返回值：某年某月有多少天
{
    ……
}
int getWeekDay(int year, int month, int day)
{
    ……
}
```

专职打印，无需返回值

专职打印，无需返回值





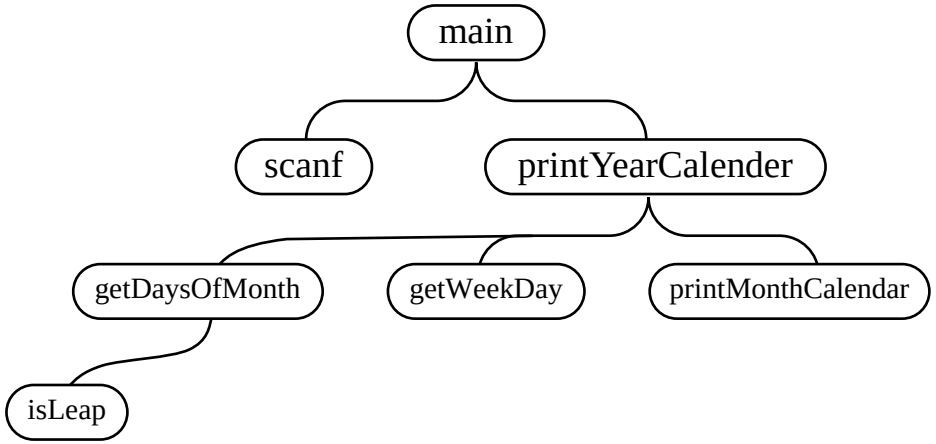
我的手机版

我的C编程版

```
month: 10
-----------------------------------------------------
Sun     Mon     Tue     Wed     Thu     Fri     Sat
-----------------------------------------------------
                1       2       3       4       5
6       7       8       9       10      11      12
13      14      15      16      17      18      19
20      21      22      23      24      25      26
27      28      29      30      31
```

例5-9：设计万年历（函数实现）

```
int getDaysOfMonth(int year, int month) //返回值：某年某月有多少天
{
    int days = 0;
    switch(month)
    {
    case 1: case 3: case 5: case 7: case 8: case 10: case 12:
        days = 31;
        break;
    case 4:  case 6:  case 9:  case 11:
        days = 30;
        break;
    case 2:
        days = isLeap(year) ? 29 : 28;
        break;
    }
    return days;
}

int isLeap(int year)        // 代码复用
{
    return (year%4 == 0) && (year%100 != 0) || (year%400 == 0);
    // 逢四则闰，百年不闰；四百再闰
}
```



每月的天数

1	2	3	4	5	6	7	8	9	10	11	12
31	28 29	31	30	31	30	31	31	30	31	30	31

例5-9：设计万年历（函数实现）

代码复用（例5-7，Zeller公式）

```
int getWeekDay(int year, int month, int day)
{
    int century, weekday;
    if(month < 3)
    {
        year--;
        month += 12;
    }
    century = year / 100;
    year %= 100;
    // Zeller formula
    weekday = (year + year/4 + century/4 - 2*century + (26*(month+1))/10 + day - 1) % 7;
    if (weekday < 0)
    {
        weekday += 7;
    }
    return weekday;
}
```

⇦ 函数重在接口！函数要实现复用！

求某年year的某月month的第一天是星期几：

```
weekday = getWeekDay(year, month, 1);
```

如，输入 2024 8 1，直接算出2024年8月1日是星期四。

直接调用已经写好的 zeller 公式，以前写的直接拿来用！

$$W = \left(\left\lfloor\frac{C}{4}\right\rfloor - 2C + Y + \left\lfloor\frac{Y}{4}\right\rfloor + \left\lfloor\frac{26(M+1)}{10}\right\rfloor + D - 1\right) \bmod 7$$

Where

W: the day of week. (0 = Sunday, 1 = Monday, ..., 5 = Friday, 6 = Saturday)

C: the zero-based century. ($= \lfloor \text{year}/100 \rfloor = \text{century} - 1$)

Y: the year of the century. ($= \begin{cases} \text{year} \bmod 100, & M = 3, 4, \ldots, 12, \\ (\text{year} - 1) \bmod 100, & M = 13, 14. \end{cases}$)

M: the month. (3 = March, 4 = April, 5 = May, ..., 14 = February)

D: the day of the month.

NOTE: In this formula January and February are counted as **months 13 and 14 of the previous year.** E.g. if it is 2010/02/02, the formula counts the date as 2009/14/02.

例5-9：设计万年历（函数实现）

- 主函数简简单单，清清爽爽。
- 每个函数实现特定功能，实现更清晰，更易于维护！
- 利用了以前写的函数，减少重复劳动，如 getWeekDay(), isLeap()等。

```
void printYearCalender(int year);                    // 打印年历
void printMonthCalendar(int month, int days, int weekday);
int getWeekDay(int year, int month, int day);  //查询某天是星期几
int isLeap(int year);   //判断是否是闰年
int getDaysOfMonth(int year, int month);        // 获得月天数

int main() // main函数清清爽爽
{
    int year, weekday;

    printf("\ninput year: ");
    scanf("%d", &year);

    printYearCalender(year); //打印年历

    return 0;
}
```

代码复用，开发省力；
单一责任，接口简洁；
意义明确，格式优雅；
调试方便，扩展便捷。