

G 唯一指定ID

前言

这题比较复杂，下面尽量说明了从头到尾的思路。
最后有完整代码，但还是建议先看思路再看代码。

分析

需要进行下列验证

1. 身份证号码长度验证
2. 每位字符是不是符合规范（1-17位数字，18位数字或X x）
3. 校验码是否正确
4. 出生年月日是否合法

验证成功后则需要计算日期间相差的天数

对于每一项要实现的功能，可以写成一个函数，方便使用，使结构更清晰。
另外函数中return关键字跳过后续内容的特性使得这样做可以更简便地处理逻辑关系。

实现

首先把主体逻辑写出来，列出需要的功能

```
（伪代码）
int main(){
    读入字符串
    判断长度是否为18，如果不是，则输出错误，结束程序
    判断每个字符是不是合法，如果不是，则输出错误，结束程序
    判断校验码是否正确，如果不是，则输出错误，结束程序
    判断日期是否合法，如果不是，则输出错误，结束程序

    都检验通过了，就计算出生到现在的天数，输出这个数字
}
```

可见，需要编写如下几个功能（函数）

1. 判断每个字符是否合法
2. 判断校验码是否正确
3. 判断日期是否合法
4. 计算出生到现在的天数

下面进行每一项功能的编写

1. 判断每个字符是否合法

```
int isCharValid(char n[]){
    int i;
    for(i = 0; i < 17; i++){
        //判断前17个字符是不是数字，如果不是，直接return 0（否），跳过之后的步骤
        if(n[i] < '0' || n[i] > '9'){
            return 0;
        }
    }
    //运行到这里，说明前面17个数字都已经判断正确了，如果前17个数有错，那么就会在之前return 0,
    if(n[17] == 'x' || n[17] == 'X' || (n[17] >= '0' && n[17] <= '9')){
        //如果第18个字符也满足了要求，就返回1
        return 1;
    }
    //不满足就返回0
    return 0;
}
```

2. 判断校验码是否正确

```
int isModValid(char n[]){
    int fac[18] = {7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2, 0}; //每位数字对应
    char check[11] = {'1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2'}; //0~10每个余数对应的
    int i, sum = 0;
    for(i = 0; i < 17; i++){
        sum += (n[i] - '0') * fac[i];
    }
    int mod = sum % 11; //求出余数
    //判断第18个字符是否和余数对应的字符相等
    return check[mod] == n[17] || (mod == 2 && n[17] == 'x');
}
```

3. 判断日期是否合法 & 4. 计算出生到现在的天数

这两条放在一起说，是因为都与日期相关，比较复杂。对于其中共性的东西，可以分离出一些通用的函数，方便计算。

下面分析实现这两个功能的过程，请注意观察注释中提出的疑问

目标1：判断日期是否合法

(伪代码)

```
int isValid(int y, int m, int d){
    如果m不在 1~12 的范围, 不合法, 返回0
    如果d不在 1~这个月天数 的范围, 不合法, 返回0 //疑问: 怎么知道这个月的天数?
    如果日期大于2019.4.2或小于1950.1.1, 不合法, 返回0 //疑问: 怎么比较两个日期的大小?

    如果上述条件都通过了, 返回1
}
```

目标2: 计算出生到现在的天数

(伪代码)

```
int countDay(int y1, int m1, int d1, int y2, int m2, int d2){
    设置一个计数用的变量 (int days = 0)
    使用while循环, 当日期y1m1d1小于y2m2d2的时候 //疑问: 怎么比较两个日期的大小?
    {
        把y1m1d1增加一天 //疑问: 怎么计算日期的加法?
        把days变量加一
    }
    现在days的值就是两个日期相隔的天数
}
```

根据上面提出的疑问, 我们发现还需实现下面几个需求:

目标3 获取一个月的天数

目标4 比较两个日期的大小

目标5 计算日期的加法

目标3: 获取一个月的天数

```
//判断是不是闰年
int isL(int y){
    return y % 400 == 0 ? 1 : (y % 100 == 0 ? 0 : y % 4 == 0);
}
//获取一个月的天数
int dayOfMonth(int y, int m){
    int md[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if(isL(y) && m == 2){
        //如果闰年二月, 就返回29
        return 29;
    }
    //否则, 就从数组md里面取出月份对应的一项, 返回
    return md[m - 1];
}
```

目标4: 比较两个日期的大小

```

//判断y1m1d1小于等于y2m2d2
int compareDate(int y1, int m1, int d1, int y2, int m2, int d2){
    if(y1 < y2){
        return 1;
    }
    if(y1 == y2){
        if(m1 < m2){
            return 1;
        }
        if(m1 == m2){
            if(d1 <= d2){
                return 1;
            }
        }
    }
    return 0;
}

```

目标5：计算日期的加法

这里用到了指针，意思是调用这个函数的时候填入三个对应变量的指针（&y, &m, &d），函数内部会运算并修改这几个指针对应的值。

类比：就像scanf函数里面填入&n，它会根据读入的内容取修改变量n的值

```

//给ymd加一天
void addDay(int *y, int *m, int *d){
    *d = *d + 1;
    if(*d > dayOfMonth(*y, *m)){//这里就调用了目标4中的函数，如果d超出，就进位到m
        *d = 1;
        *m = *m + 1;
        if(*m > 12){//如果m超出，就进位到y
            *m = 1;
            *y = *y + 1;
        }
    }
}

```

现在解决了目标3、4、5，就可以回到目标1、2的伪代码里，把之前疑问的地方用新写的函数补上

```

int isDateValid(int y, int m, int d){

    //如果m不在 1~12 的范围，不合法，返回0
    if(m < 1 || m > 12){
        return 0;
    }
    //如果d不在 1~这个月天数 的范围，不合法，返回0
    if(d < 1 || d > dayOfMonth(y, m)){
        return 0;
    }
    //如果日期大于2019.4.2或小于1950.1.1，不合法，返回0
    if(y < 1950 || y > 2019){
        return 0;
    }
    if(!compareDate(y, m, d, 2019, 4, 2)){
        return 0;
    }
    //如果上述条件都通过了，返回1
    return 1;
}

int countDay(int y1, int m1, int d1, int y2, int m2, int d2){
    //设置一个计数用的变量
    int days = 0;
    //使用while循环，当日期y1m1d1小于y2m2d2的时候
    while(compareDate(y1, m1, d1, y2, m2, d2)){
        addDay(&y1, &m1, &d1); //把y1m1d1增加一天
        days++; //把days变量加一
    }
    return days-1;
}

```

终

现在目标1、2、3、4、5都实现了，回到main函数里，把之前的伪代码补全

```

int main(){
    char n[10000];
    char *fail = "Chawuciren";

    //读入字符串
    scanf("%s", &n);
    //判断长度是否为18, 如果不是, 则输出错误, 结束程序
    if(strlen(n) != 18){
        printf(fail);
        return 0;
    }
    //判断每个字符是不是合法, 如果不是, 则输出错误, 结束程序
    if(!isCharValid(n)){
        printf(fail);
        return 0;
    }
    //判断校验码是否正确, 如果不是, 则输出错误, 结束程序
    if(!isModValid(n)){
        printf(fail);
        return 0;
    }
    //判断日期是否合法, 如果不是, 则输出错误, 结束程序
    int year = (n[6] - '0') * 1000 + (n[7] - '0') * 100 + (n[8] - '0') * 10 + (n[9] - '0');
    int month = (n[10] - '0') * 10 + (n[11] - '0');
    int day = (n[12] - '0') * 10 + (n[13] - '0');
    if(!isDateValid(year, month, day)){
        printf(fail);
        return 0;
    }

    //都检验通过了, 就计算出出生到现在的天数, 输出这个数字
    printf("%d", countDay(year, month, day, 2019, 4, 2));
}

```

完整代码

```

#include <stdio.h>
#include <string.h>

int isModValid(char n[]){
    int fac[18] = {7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2,0};
    char check[11] = {'1','0','X','9','8','7','6','5','4','3','2'};
    int i, sum = 0;
    for(i = 0; i < 17; i++){
        sum += (n[i] - '0') * fac[i];
    }
    int mod = sum % 11;
    return check[mod] == n[17] || (mod == 2 && n[17] == 'x');
}

int isCharValid(char n[]){
    int i;
    for(i = 0; i < 17; i++){
        if(n[i] < '0' || n[i] > '9'){
            return 0;
        }
    }
    if(n[17] == 'x' || n[17] == 'X' || (n[17] >= '0' && n[17] <= '9')){
        return 1;
    }
    return 0;
}

int isL(int y){
    return y % 400 == 0 ? 1 : (y % 100 == 0 ? 0 : y % 4 == 0);
}

int dayOfMonth(int y, int m){
    int md[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if(isL(y) && m == 2){
        return 29;
    }
    return md[m - 1];
}

int isDateValid(int y, int m, int d){

    if(m < 1 || m > 12){
        return 0;
    }
    if(d < 1 || d > dayOfMonth(y, m)){
        return 0;
    }
    if(y < 1950 || y > 2019){
        return 0;
    }
    if(y == 2019){
        if(m > 4){
            return 0;
        }
    }
}

```

```

        if(m == 4){
            if(d > 2){
                return 0;
            }
        }
    }
    return 1;
}

int compareDate(int y1, int m1, int d1, int y2, int m2, int d2){
    if(y1 < y2){
        return 1;
    }
    if(y1 == y2){
        if(m1 < m2){
            return 1;
        }
        if(m1 == m2){
            if(d1 <= d2){
                return 1;
            }
        }
    }
    return 0;
}

void addDay(int *y, int *m, int *d){
    *d = *d + 1;
    if(*d > dayOfMonth(*y, *m)){
        *d = 1;
        *m = *m + 1;
        if(*m > 12){
            *m = 1;
            *y = *y + 1;
        }
    }
}

int countDay(int y1, int m1, int d1, int y2, int m2, int d2){
    int days = 0;
    while(compareDate(y1, m1, d1, y2, m2, d2)){
        addDay(&y1, &m1, &d1);
        days++;
    }
    return days-1;
}

int main(){
    char n[10000];
    char *fail = "Chawuciren";

    scanf("%s", &n);
    if(strlen(n) != 18){
        printf(fail);
    }
}

```



```
        return 0;
    }
    if(!isCharValid(n)){
        printf(fail);
        return 0;
    }
    if(!isModValid(n)){
        printf(fail);
        return 0;
    }
    int year = (n[6] - '0') * 1000 + (n[7] - '0') * 100 + (n[8] - '0') * 10 + (n[9] - '0');
    int month = (n[10] - '0') * 10 + (n[11] - '0');
    int day = (n[12] - '0') * 10 + (n[13] - '0');
    if(!isDateValid(year, month, day)){
        printf(fail);
        return 0;
    }
    printf("%d", countDay(year, month, day, 2019, 4, 2));
}
```

最后

这一题的实现过程中，出现了一些日期运算相关的东西。这些东西很有可能在以后做题的时候还会用到（比如说日期的推算），所以可以把相关函数保存起来，下次再用的时候，直接拿出来用

by lxr