

# 南京理工大学计算机科学与工程学院

## 程序设计基础（Ⅱ）实验报告

班    级\_\_\_\_\_9181069502\_\_\_\_\_

学生姓名\_\_\_\_\_黄海浪\_\_\_\_\_

学    号\_\_\_\_\_9181040G0818\_\_\_\_\_

起止时间\_\_\_\_\_2020.08.11-2020.08.12\_\_\_\_\_

教    师\_\_\_\_\_刘冬梅\_\_\_\_\_

## 目录

一、	问题描述与实验要求.....	3
二、	实验设计思想与分析.....	3
三、	函数流程图 .....	4
四、	伪代码描述 .....	9
五、	寄存器使用对照表 .....	10
六、	程序结果.....	12
七、	调试记录与心得体会 .....	15
	附录 MIPS 代码.....	16

## 一、问题描述与实验要求

### 简单要求:

实现年历，有完整输入输出、有函数调用（寄存器/栈）。

### 分析:

前面两个题目课后已经写过了，于是选择了实现年历，年历实现 3 个功能，1. 显示某年某月的月历，2. 显示某年整年的年历，3. 输入两个日期计算天数差。实现了函数调用，完整的输入输出，循环控制以及模块化、函数式编程。

## 二、实验设计思想与分析

### 设计思想（模块化、函数式设计）:

输入输出：从实验来看，涉及到的输入输出比较多，于是将输入输出写成函数以实现字符串、整数的输入输出（虽然结果表明用不到字符串的输入），分别为 `coutStr`, `coutInt`, `cinStr`, `cinInt`。

数组操作：实验上涉及数组读取比较多，因为要计算每个月天数，所以单独提出来写成一个函数 `monthArrInd`。修改因只需要修改一个地方且只需要一行代码 `sw $x, i*4(lable)`，所以并没有单独写成函数。

3 个功能：显示月历、显示年历、日期差三个功能分别单独提出来作为三个函数，分别为：`calMonth`、`calYear`、`calDate2Date`。

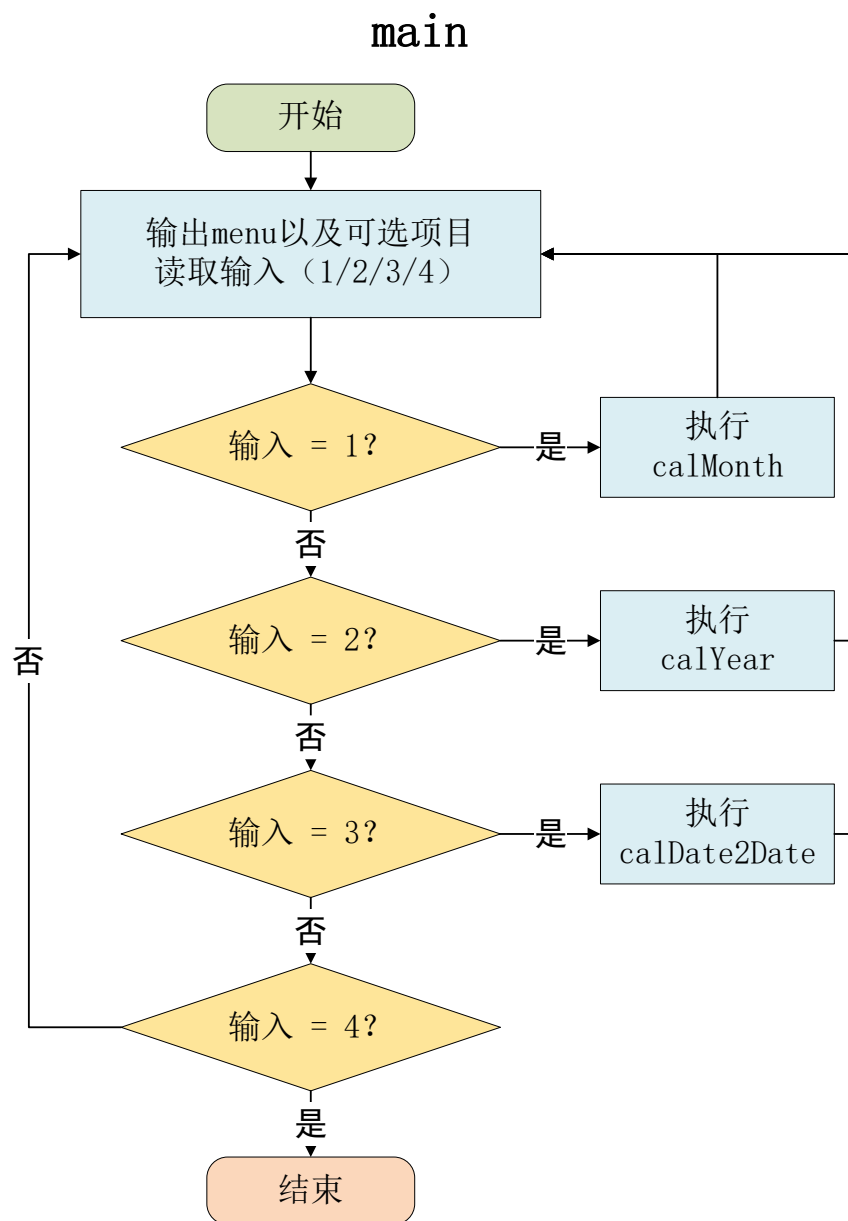
其他：因为月历、年历有共性，所以把具体到某个月的计算与输出（仅计算部分与输出部分）单独提出来作为一个函数为：`calCertainMonth`。又因为月历、年历以及日期差三个功能对闰年判断次数多，所以也单独提出来作为一个函数：`isLeapYear`。

### 要求体现:

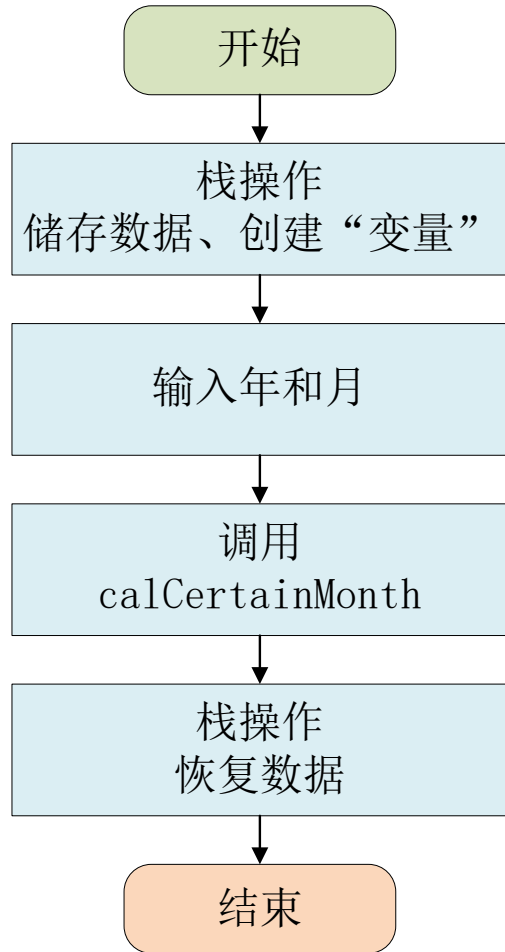
完整的输入输出：体现在输入输出的模块化，且使用了较多的 `.data` 实现完整的提示，输入提供两个缓冲对，一个是 `TEMPSTR`，`TEMPINT`。

栈的使用：函数调用基本以 `a0-a4` 作为参数传递，由于参数过少并未使用栈进行传递，但是使用了栈对输入参数的保存，输出参数的保存，`$ra` 的保存以及上述参数的恢复。栈操作还进行临时保护变量创建，因为要使用被调用者保存的变量，但是写在最前会增加阅读难度，所以用栈进行了临时变量的创建。

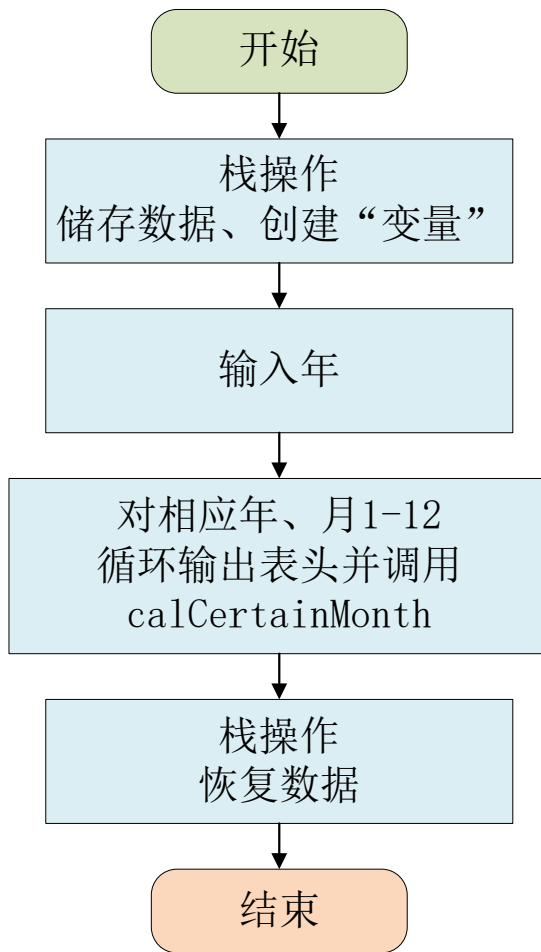
### 三、函数流程图



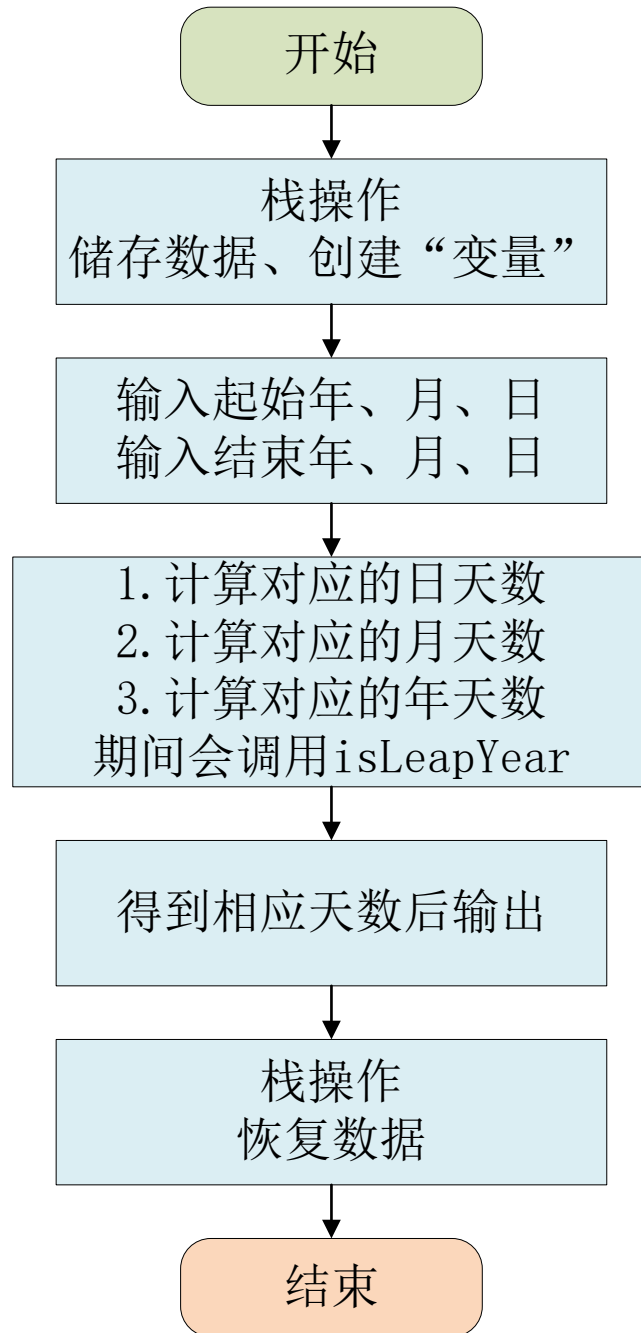
## calMonth



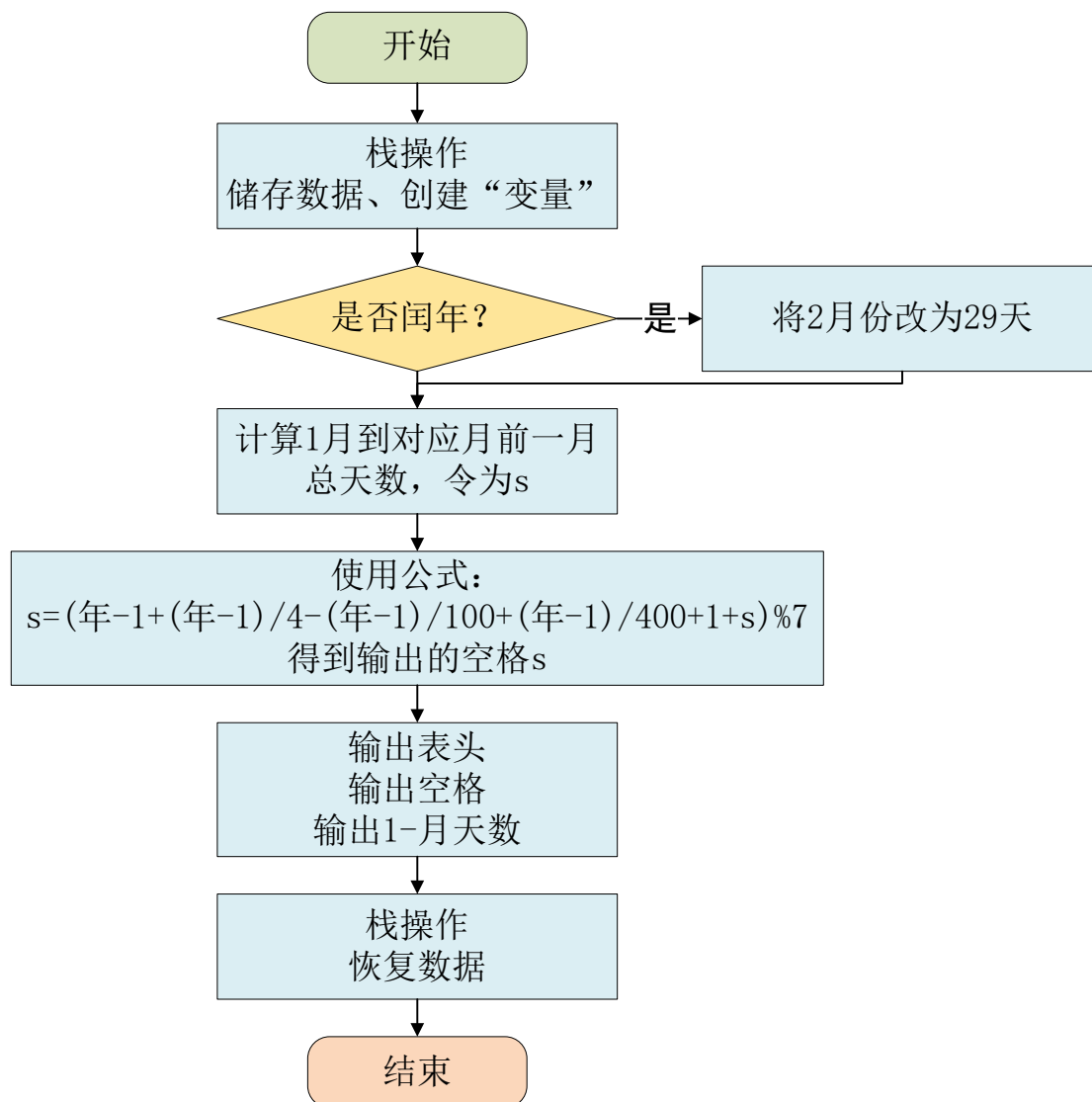
## calYear



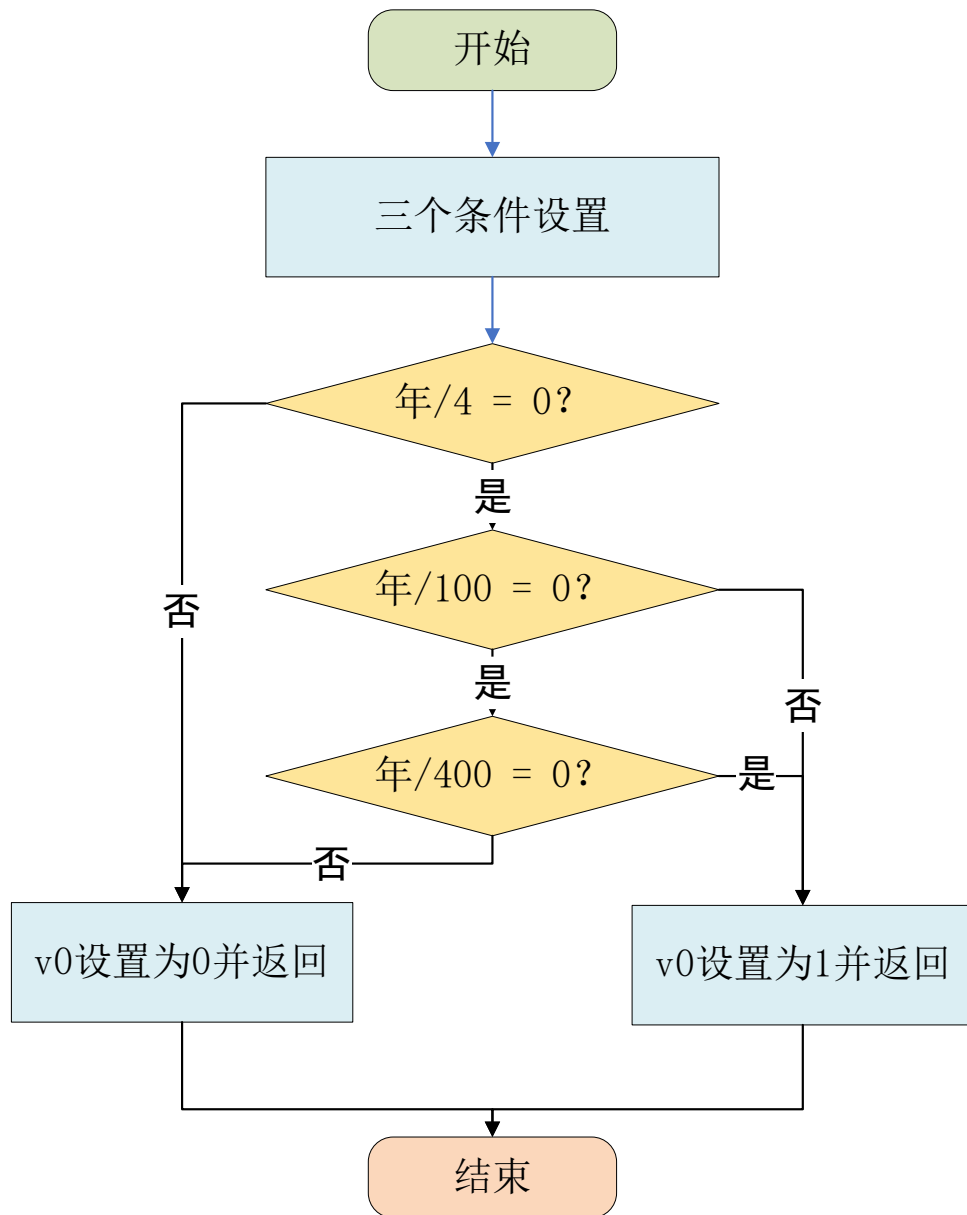
## calDate2Date



## calCertainMonth



# isLeapYear





## 四、伪代码描述

因为输入输出代码、闰年判断等都比较简单，下面给出部分功能得 c++实现：

下面是 mips 中 calCertainMonth 的实现：

```
void week(int year, int month)    //求出日期对应的星期函数
{
    int arr[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    if (((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0)) {
        arr[2] = 29;
    }
    int s = 0;
    for (int i = 1; i < month; ++i) {
        s += arr[i];
    }
    s = (year - 1 + (year - 1) / 4 - (year - 1) / 100 + (year - 1) / 400 + 1 + s) % 7;    //所求日期的星期数
    cout << setw(6) << "日" << setw(6) << "一" << setw(6) << "二"
    << setw(6) << "三" << setw(6) << "四" << setw(6) << "五"
    << setw(6) << "六" << endl;

    for (int j = 0; j < s; j++){
        cout << setw(6) << "";
    }

    for (int t = 1; t <= arr[month]; t++){
        cout << setw(6) << t;
        s = (s + 1) % 7;
        if (s % 7 == 0)
        {
            cout << endl;
        }
    }
}
```

下面是 mips 中 calDate2Date 的实现：

```
void calDay(int sYear, int sMonth, int sDay,
            int eYear, int eMonth, int eDay){

    int sum = 0;
    sum -= sDay;
    sum += eDay;
```

```

    monthArr[2] = 28;
    if (isLeapYear(sYear)) monthArr[2] = 29;
    for (int i=1;i<sMonth;++i) sum -= monthArr[i];

    monthArr[2] = 28;
    if (isLeapYear(eYear)) monthArr[2] = 29;
    for (int i=1;i<eMonth;++i) sum += monthArr[i];

    while (sYear<eYear) {
        sum += 365;
        if(isLeapYear(sYear)) ++sum;
        ++sYear;
    }
    cout<<sum<<endl;
}

```

## 五、寄存器使用对照表

输入函数			输出函数		
cinStr	输入参数	无	coutStr	输入参数 str 地址	a0
	返回值	TEMPSTR		返回值	无
cinInt	输入参数	无	cinInt	输入参数 Int 值	a0
	返回值	TEMPINT		返回值	无

monthArrInd		isLeapYear	
输入 下标 int	a0	输入 Year int	a0
MONTHARR 地址	t0、t1	中间判断参数	t0、t1、t2
返回 天数 int	v0	返回 闰年 1 平年 0	v0

calCertainMonth	
输入 Year Month	a0、a1
year	a0、s0
month	a1、s1
中间变量 29(改变 2 月)、计算周、for 里面的 i 其他	t3、s2、s3 t5 等
返回	无

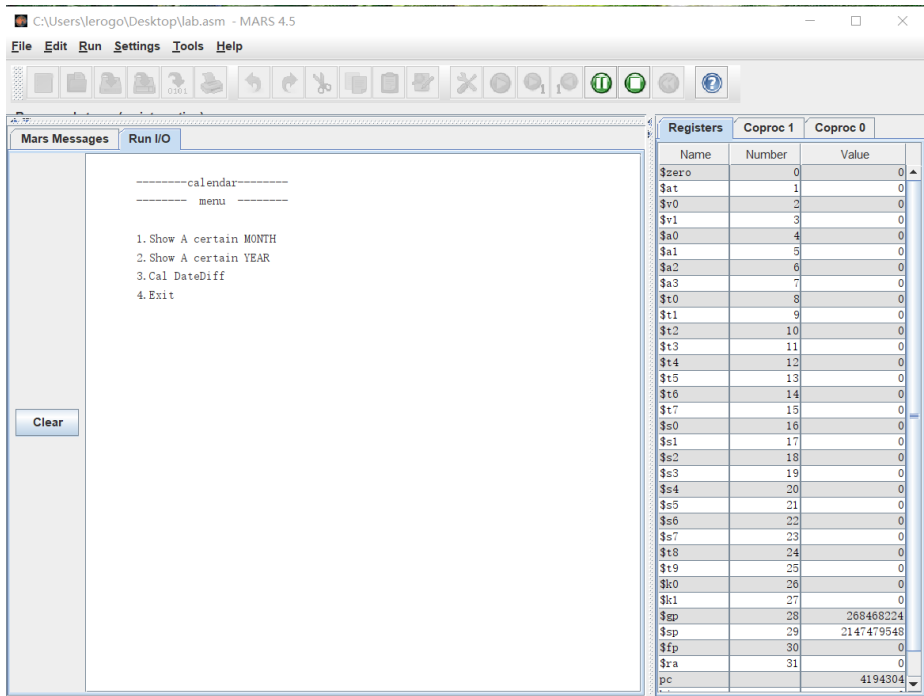
calMonth	
输入	无
year	s0
month	s1
返回	无

calYear	
输入	无
year	s0
Month (1-12)	s1
返回	无

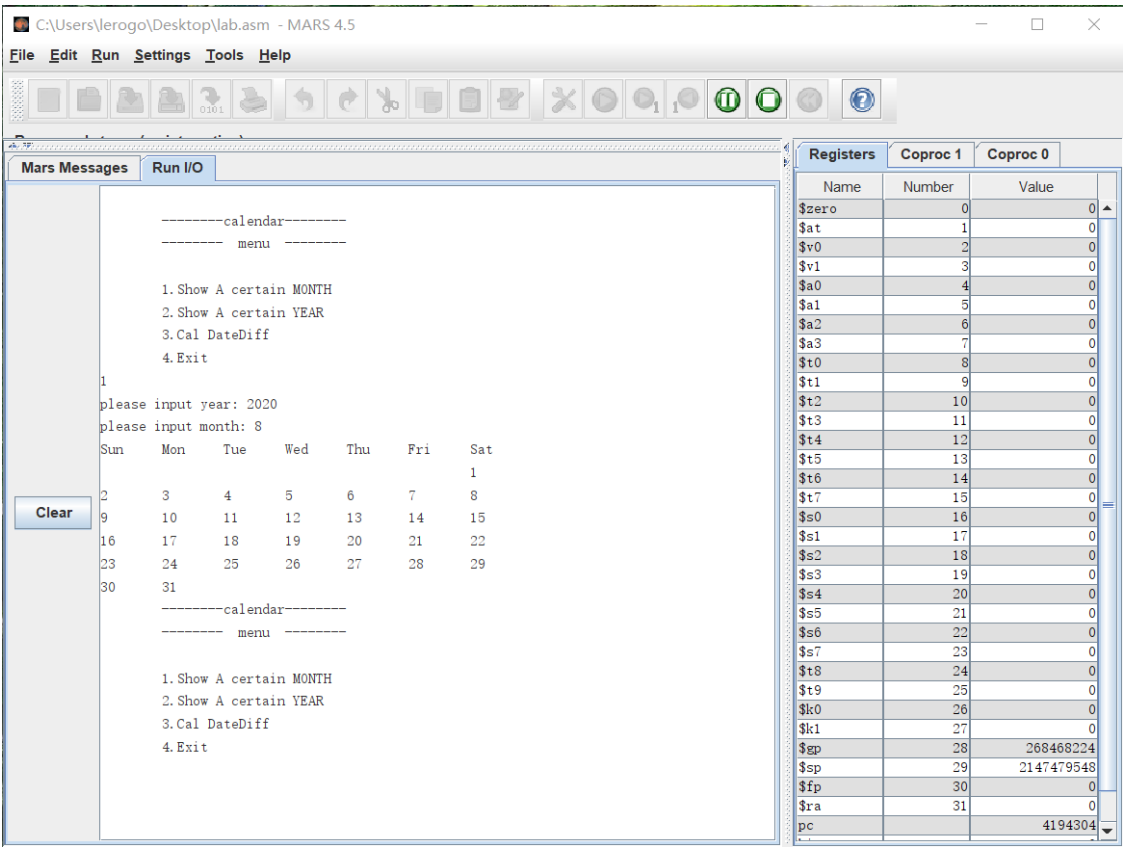
calDate2Date	
输入	无
起始 year	s1
起始 month	s2
起始 day	s3
结束 year	s4
结束 month	s5
结束 day	s6
总时间天数	S7
返回	无

## 六、程序结果

程序可以通过代码证明没有错误，多次尝试过，以下放一个随机的测试截图  
初始运行



测试 1 某年某月 月历



## 测试 2 某年年历

C:\Users\lerogo\Desktop\lab.asm - MARS 4.5

File Edit Run Settings Tools Help

Mars Messages Run I/O

```

-----calendar-----
----- menu -----

1. Show A certain MONTH
2. Show A certain YEAR
3. Cal DateDiff
4. Exit

2
please input year: 2020

        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

```

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194304

## 测试 3 天数计算

C:\Users\lerogo\Desktop\lab.asm - MARS 4.5

File Edit Run Settings Tools Help

Mars Messages Run I/O

```

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

-----calendar-----
----- menu -----

1. Show A certain MONTH
2. Show A certain YEAR
3. Cal DateDiff
4. Exit

3
please input start year: 1973
please input start month: 6
please input start day: 5
please input end year: 2020
please input end month: 4
please input end day: 3

the DateDiff is: 17104

-----calendar-----
----- menu -----

1. Show A certain MONTH
2. Show A certain YEAR
3. Cal DateDiff
4. Exit

```

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194304

### 测试 3 对比结果正确

首页 > 日期差计算

 日期差计算

开始时间: 1973 年 06 月 05 日 00 : 00 : 00  
结束时间: 2020 年 04 月 03 日 00 : 00 : 00


计算两个日期相差天数

两个时间相差	17104天 0时 0分 0秒
按天计算相差	17104天
按时计算相差	410496时
按分计算相差	24629760分
按秒计算相差	1477785600秒

### 测试 4 退出程序

C:\Users\Verogo\Desktop\lab.asm - MARS 4.5

File Edit Run Settings Tools Help



Mars Messages

Run I/O

27 28 29 30 31

-----calendar-----  
----- menu -----  
  
1. Show A certain MONTH  
2. Show A certain YEAR  
3. Cal DateDiff  
4. Exit  
  
3  
please input start year: 1973  
please input start month: 6  
please input start day: 5  
please input end year: 2020  
please input end month: 4  
please input end day: 3  
  
the DateDiff is: 17104  
-----calendar-----  
----- menu -----  
  
1. Show A certain MONTH  
2. Show A certain YEAR  
3. Cal DateDiff  
4. Exit  
  
4  
-- program is finished running --

Clear

Registers

Coproc 1

Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	4
\$a1	5	12
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	19
\$t2	10	19
\$t3	11	29
\$t4	12	0
\$t5	13	31
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	4194320
pc		4194392

## 七、调试记录与心得体会

这个程序因为写了伪代码，而且多次测试过，也从代码逻辑去验证它就是正确的，截很多图显得比较冗余，代码 copy 下来就可以运行了。

比较难的部分是月历输出，因为要涉及到周得计算，还涉及诸多变量等，以及输出得格式化。期间调用 `calCertainMonth` 时把 `v0` 当作 `a0` 带入参数，得到计算结果不正确。还有就是函数的写法，必须使用 `jal` 进行函数调用，而不是无条件跳转（误用导致了不断往复的死循环）。本次实验只出现了上面两次需要 debug 难找到的错误，至于小错误基本没有。

比较烦的部分是日期计算，因为涉及到的变量比较多，但是好在计算日之后存放日的变量可以不再使用，计算月之后存放月的变量也不再使用，所以可以让其作为临时变量，不用再去操作栈，但是有多个日期计算，就很烦躁，需要细心。

这个报告完成于快开学之际，由于前面两个题目之前就做过了，所以选择了一个没有做过的题目，目的是复习汇编之后，能够更加清晰地记住这些命令以及熟悉汇编。做来做去感觉并不是很难，主要是需要熟悉汇编命令，记住 `int` 用 `word`，`str` 用 `space` 缓存，`align` 用来对齐。栈的使用在脑子里模拟栈的结构，`-x` 就是往下移动 `x`，然后 `offset` 就是往上走一部分，储存的数据就是向上走。记住这些和命令，基本上写程序没什么问题。

写一个类似于 `c++` 的结构，需要在 `main` 进行 `syscall` 退出操作，调用函数需要 `jal` 链接，关于循环结构则使用命名格式（在后面加上大范围标签的后缀名），这样进行操作不会出现重名，循环结构使用 `b` 进行跳转。至于条件要善于使用 `mips` 提供的宏命令 `blt`, `bgt` 等等，当然自己也要会用基础的命令写出那种条件，比如用 `add\sub` 配合 `beq\bne\blez` 等等，使用 `xor` 异或进行快速判断是否相等，能够提高程序效率（量大才能感受到）。

总的来说，汇编都是熟悉的问题，所以接下来到考试的时间要多练，多看书上教材和老师讲过的代码与习题。

## 附录 MIPS 代码

代码下载: <https://cloud.lerogo.com/#/s/gPHX?path=%2F>

```
.data
STRMENU: .asciiz "\n\t-----calendar-----\n\t----- menu
-----\n\n\t1.Show A certain MONTH\n\t2.Show A certain
YEAR\n\t3.Cal DateDiff\n\t4.Exit\n"
STRWEEK: .asciiz "Sun\tMon\tTue\tWed\tThu\tFri\tSat\n"
STRINFOYEAR: .asciiz "please input year: "
STRINFOMONTH: .asciiz "please input month: "
# 计算日期差距用
STRINFOSTARTYEAR:.asciiz "please input start year: "
STRINFOSTARTMONTH: .asciiz "please input start month: "
STRINFOSTARTDAY: .asciiz "please input start day: "
STRINFOENDYEAR: .asciiz "please input end year: "
STRINFOENDMONTH: .asciiz "please input end month: "
STRINFOENDDAY: .asciiz "please input end day: "
STRINFODATEDIFF: .asciiz "\nthe DateDiff is: "
# 格式化输出
STRSPACETAB: .asciiz " \t"
STRSPACE: .asciiz " "
STRTAB: .asciiz "\t"
STRENTER: .asciiz "\n"
STRMONTH: .asciiz "\n\n\t\t Month: "
# 每个月天数 初始为平年
MONTHARR: .align 2
.word 0,31,28,31,30,31,30,31,31,30,31,30,31
# 输入缓冲
TEMPSTR: .space 100
TEMPINT: .align 2
.space 80

.text
.globl main
main:
# 显示菜单
la $a0,STRMENU
jal coutStr
# 输入选项
jal cinInt
lw $a0,TEMPINT
```



# 判断输入 输入 1 则显示月历，2 则显示年历，3 计算日期差，4 退出，其他则重新开始 main

```
xor $t0,$a0,1
bnez $t0,notCalMonth_main
    jal calMonth
    b main
notCalMonth_main:
xor $t0,$a0,2
bnez $t0,notCalYear_main
    jal calYear
    b main
notCalYear_main:
xor $t0,$a0,3
bnez $t0,notcalDate2Date_main
    jal calDate2Date
    b main
notcalDate2Date_main:
xor $t0,$a0,4
bnez $t0,main

li $v0,10
syscall
```

calMonth:

```
# 储存 ra s0 s1
addiu $sp,$sp,-12
sw $ra,($sp)
sw $s0,4($sp)
sw $s1,8($sp)
# 输出 please input year:
la $a0,STRINFOYEAR
jal coutStr
jal cinInt # 输入年
lw $s0,TEMPINT # s0:年
# 输出 please input month:
la $a0,STRINFOMONTH
jal coutStr
jal cinInt # 输入月
lw $s1,TEMPINT # s1:月

move $a0,$s0
move $a1,$s1
jal calCertainMonth # 计算确定的某年某月
```

```

lw $ra, ($sp)
lw $s0, 4($sp)
lw $s1, 8($sp)
addiu $sp, $sp, 12
move $t0, $zero
jr $ra

```

calYear:

```

addiu $sp, $sp, -12
sw $ra, ($sp)
sw $s0, 4($sp)
sw $s1, 8($sp)

```

# 输出 please input year:

```

la $a0, STRINFOYEAR
jal coutStr
jal cinInt    # 输入年
lw $s0, TEMPINT    # s0:年

```

```
li $s1, 1
```

loopfor1\_calYear:

```
    bgt $s1, 12, end_loopfor1_calYear
```

# 输出表头 Month:x

```

la $a0, STRMONTH
jal coutStr
move $a0, $s1
jal coutInt
la $a0, STRENTER
jal coutStr

```

```
move $a0, $s0
```

```
move $a1, $s1
```

```
jal calCertainMonth    # 计算确定的某年某月
```

```
addi $s1, $s1, 1
```

```
b loopfor1_calYear
```

end\_loopfor1\_calYear:

```
lw $ra, ($sp)
```

```
lw $s0, 4($sp)
```

```
lw $s1, 8($sp)
```

```
addiu $sp, $sp, 12
```

```
jr $ra
```

```

calDate2Date:
    # 储存 ra s1-s6
    addiu $sp, $sp, -32
    sw $ra, ($sp)
    sw $s1, 4($sp)
    sw $s2, 8($sp)
    sw $s3, 12($sp)
    sw $s4, 16($sp)
    sw $s5, 20($sp)
    sw $s6, 24($sp)
    sw $s7, 28($sp)

    # 输出 please input start year:
    la $a0, STRINFOSTARTYEAR
    jal coutStr
    jal cinInt    # 输入年
    lw $s1, TEMPINT    # s1:年 start
    # 输出 please input start month:
    la $a0, STRINFOSTARTMONTH
    jal coutStr
    jal cinInt    # 输入月
    lw $s2, TEMPINT    # s2:月 start
    # 输出 please input start day:
    la $a0, STRINFOSTARTDAY
    jal coutStr
    jal cinInt    # 输入日
    lw $s3, TEMPINT    # s3:日 start

    # 输出 please input end year:
    la $a0, STRINFOENDYEAR
    jal coutStr
    jal cinInt    # 输入年
    lw $s4, TEMPINT    # s4:年 end
    # 输出 please input end month:
    la $a0, STRINFOENDMONTH
    jal coutStr
    jal cinInt    # 输入月
    lw $s5, TEMPINT    # s5:月 end
    # 输出 please input end day:
    la $a0, STRINFOENDDAY
    jal coutStr
    jal cinInt    # 输入如
    lw $s6, TEMPINT    # s6:日 end

```

```

li $s7, 0
add $s7, $s7, $s6
sub $s7, $s7, $s3
# 计算 start 月时间
move $a0, $s1
jal isLeapYear
li $t3, 28
sw $t3, MONTHARR+8
beqz $v0, notleap_start_month_calDate2Date
    li $t3, 29
    sw $t3, MONTHARR+8
notleap_start_month_calDate2Date:
li $s3, 1
loop_start_month_calDate2Date:
    ble $s2, $s3, end_loop_start_month_calDate2Date
    move $a0, $s3
    jal monthArrInd
    sub $s7, $s7, $v0
    addi $s3, $s3, 1
    b loop_start_month_calDate2Date
end_loop_start_month_calDate2Date:
# 计算 end 月时间
move $a0, $s4
jal isLeapYear
li $t3, 28
sw $t3, MONTHARR+8
beqz $v0, notleap_end_month_calDate2Date
    li $t3, 29
    sw $t3, MONTHARR+8
notleap_end_month_calDate2Date:
li $s6, 1
loop_end_month_calDate2Date:
    ble $s5, $s6, end_loop_end_month_calDate2Date
    move $a0, $s6
    jal monthArrInd
    add $s7, $s7, $v0
    addi $s6, $s6, 1
    b loop_end_month_calDate2Date
end_loop_end_month_calDate2Date:

# 计算年间隔时间
loop_year_calDate2Date:
    slt $t0, $s1, $s4

```

```

        beqz    $t0, end_loop_year_calDate2Date
        addi    $s7, $s7, 365
        move    $a0, $s1
        jal isLeapYear
        beqz    $v0, notleap_year_calDate2Date
            addi    $s7, $s7, 1
        notleap_year_calDate2Date:
        addi    $s1, $s1, 1
        b      loop_year_calDate2Date
end_loop_year_calDate2Date:
la  $a0, STRINFODATEDIFF
jal coutStr
move    $a0, $s7
jal coutInt

```

```

lw  $ra, ($sp)
lw  $s1, 4($sp)
lw  $s2, 8($sp)
lw  $s3, 12($sp)
lw  $s4, 16($sp)
lw  $s5, 20($sp)
lw  $s6, 24($sp)
lw  $s7, 28($sp)
addiu $sp, $sp, 32
jr  $ra

```

```

isLeapYear:
# 判断是不是闰年 是 返回 v0 = 1
rem $t0, $a0, 4
rem $t1, $a0, 100
rem $t2, $a0, 400
bnez    $t0, continue400_isLeapYear
        beqz    $t1, continue400_isLeapYear
            li    $v0, 1
            jr    $ra
continue400_isLeapYear:
        bnez    $t2, notleap_isLeapYear
            li    $v0, 1
            jr    $ra
notleap_isLeapYear:
        li    $v0, 0
        jr    $ra

```

calCertainMonth: #a0:year a1:month

```

addiu $sp,$sp,-12
sw $ra,($sp)
sw $s0,4($sp)
sw $s1,8($sp)

move $s0,$a0      # s0:year
move $s1,$a1      # s1:month

li $t3,29
jal isLeapYear
beqz $v0,notleap_calCertainMonth
    sw $t3,MONTHARR+8
notleap_calCertainMonth:
# 将会使用 s2 计算周 s3 作为 i 储存原来的 s2/s3
addiu $sp,$sp,-8
sw $s2,($sp)
sw $s3,4($sp)

li $s3,1          #s3:for i = 1
move $s2,$zero
loopfor1_calCertainMonth:
    ble $s1,$s3,end_loopfor1_calCertainMonth #判断条件 i < month
    move $a0,$s3          # 准备计算 某月多少天
    jal monthArrInd
    add $s2,$s2,$v0      # 加上计算结果
    addi $s3,$s3,1      # i++
    b loopfor1_calCertainMonth
end_loopfor1_calCertainMonth:
# 计算周 计算公式:  $s = (\text{year} - 1 + (\text{year} - 1) / 4 - (\text{year} - 1) / 100 + (\text{year} - 1) / 400 + 1 + s) \% 7$ 
addi $t0,$s0,-1 #t0: year - 1
div $t1,$t0,4 #t1: (year - 1) / 4
div $t2,$t0,100 #t2: (year - 1) / 100
div $t3,$t0,400 #t3: (year - 1) / 400
# s2 加上前面的
addi $s2,$s2,1 # s2 = s2 + 1
add $s2,$s2,$t0 # s2 = s2 + year - 1
add $s2,$s2,$t1 # s2 = s2 + (year - 1) / 4
sub $s2,$s2,$t2 # s2 = s2 - (year - 1) / 100
add $s2,$s2,$t3 # s2 = s2 + (year - 1) / 400
rem $s2,$s2,7 # 取余数

# 输出表头
la $a0,STRWEEK

```

```

jal coutStr
# 输出空格
li $s3,0      #t1 : i = 0
loopfor2_calCertainMonth:
    ble $s2,$s3,end_loopfor2_calCertainMonth
    la $a0,STRSPACETAB #格式化输出
    jal coutStr
    addi $s3,$s3,1
    b loopfor2_calCertainMonth
end_loopfor2_calCertainMonth:
li $s3,1      # s3:i = 1
move $a0,$s1   # 准备计算 某月多少天
jal monthArrInd
move $t5,$v0
loopfor3_calCertainMonth:    #for (int t = 1; t <= arr[month];
t++)
    blt $t5,$s3,end_loopfor3_calCertainMonth
    # 输出 日
    move $a0,$s3
    jal coutInt
    la $a0,STRSPACE #格式化输出
    jal coutStr
    la $a0,STRTAB #格式化输出
    jal coutStr

    # 格式化 输出 换行 \n      if (s % 7 == 0)
    addi $s2,$s2,1
    rem $s2,$s2,7
    bnez $s2,s_not_zero_loopfor3_calCertainMonth
        la $a0,STRENTER
        jal coutStr
    s_not_zero_loopfor3_calCertainMonth:
    addi $s3,$s3,1
    b loopfor3_calCertainMonth
end_loopfor3_calCertainMonth:
lw $s2,($sp)
lw $s3,4($sp)
addiu $sp,$sp,8
lw $ra,($sp)
lw $s0,4($sp)
lw $s1,8($sp)
addiu $sp,$sp,12
jr $ra

```

monthArrInd: # a0 为下标 从 0 开始, 输入 1 则代表 1 月 返回天数 v0

```
la $t0, MONTHARR
move $t1, $a0
loop_monthArrInd:
    beqz $t1, endloop_monthArrInd
    addi $t1, $t1, -1
    addi $t0, $t0, 4
    b loop_monthArrInd
endloop_monthArrInd:
lw $v0, ($t0)
jr $ra
```

cinStr: # 输入一个 str, 并在 TEMPSTR 返回

```
li $v0, 8
la $a0, TEMPSTR
li $a1, 100
syscall
jr $ra
```

cinInt: # 输入一个数字, 并在 TEMPINT 返回

```
li $v0, 5
syscall
sw $v0, TEMPINT
jr $ra
```

coutStr: # 将 str 地址放入 \$a0 中

```
li $v0, 4
syscall
jr $ra
```

coutInt: # 将 Int 放入 \$a0 中

```
li $v0, 1
syscall
jr $ra
```