软件测试基础与实践

实验报告

实验名称： 白盒测试实验一

实验地点： 计算机楼 268

实验日期： 2018/10/25

学生姓名： 张睦婕

学生学号： 71117133

东南大学 软件学院 制

目录

软件测试基础与实践 1
实验报告 1
东南大学 软件学院 制 1
一、实验目的 3
二、实验内容 3
1. 问题 3
2. 要求 3
3. 回答 4
I. getWeekday()的程序流程图 4
II. 语句覆盖和判定覆盖测试用例 6
III. 条件覆盖、判定条件覆盖和条件组合覆盖测试用例 8
IV. 对循环的测试用例 11
V. 路径覆盖 12
VI. MC/DC 测试用例 12
VII. 条件短路对控制流测试的影响 14
三、实验体会 14
1. 对白盒测试的理解 14
2. 程序中可能存在的错误 15
3. 对软件测试的理解 16

一、实验目的

（1）巩固基于控制流白盒测试知识，对于给定的待测程序，能熟练应用基本控制流覆盖方法设计测试用例；

（2）通过绘制程序控制流程图，实现对程序源代码的逻辑描述；

（3）掌握逻辑短路对测试的影响；

（4）培养严谨和系统的测试精神，学习测试用例的设计和分析。

二、实验内容

1. 问题

运用基于控制流的动态白盒测试方法，对 WeekA 程序中的方法 getWeekday()进行测试。设计测试用例时，尽可 能设计最少的测试用例数，同时保证每种覆盖方法的覆盖率尽可能达到 100%。

2. 要求

(1)给出 getWeekday()的程序流程图，这是进行基于控制流动态白盒测试的基础。

(2)分别以语句覆盖和判定覆盖方法设计测试用例，并写出每个测试用例的执行路径。

(3)自行写一个小程序，验证当判定中包含多个条件时，条件短路对控制流测试的影响。 通过这段小程序的执行，加强对逻辑短路现象的理解。

(4)分别以条件覆盖、判定条件覆盖和条件组合覆盖方法设计测试用例，并写出每个测试用例的执行

路径。

(5)给出对程序中循环的测试用例，并说明测试用例设计的理由。

(6)如果要进一步用路径覆盖准则来测试 getWeekday()，请基于程序流程图计算其中可能的路径条共有多少条？ 是否包含不可达路径？依照你设计测试用例的速度，完成所有路径的测试需要多少时间？

注意： A: 正确分析程序可能的执行路径； B: 对于涉及循环的路径，统计时可简化为：执行 N>1 次视为同一条路径，执行 0 次（即跳过）视为另一条 路径；

(7)给出 MC/DC(修订的判定条件覆盖)方法对下列 2 处语句的测试用例。

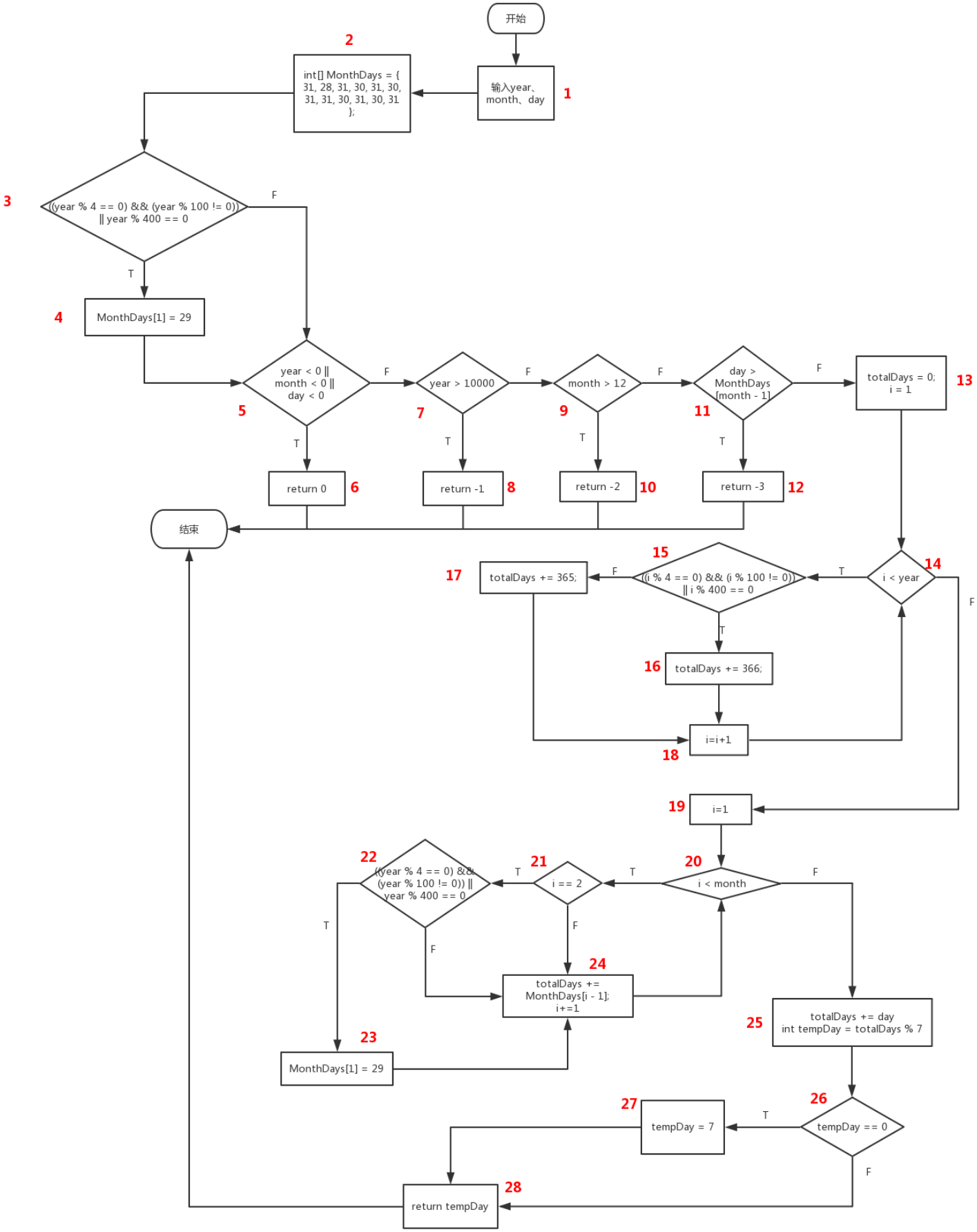
```
…
int[] MonthDays = { 31, 28, 31, 30, 31, 30, 31,
31, 30, 31, 30, 31 };
if (((year % 4 == 0) && (year % 100 != 0)) ||
year % 400 == 0) {
        MonthDays[1] = 29;
}
…
```

```
…
if (((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0) {
        totalDays += 366;
} else {
        totalDays += 365;
}
…
```

3. 回答

I. getWeekday()的程序流程图

下图如果看不清楚，请见压缩包内 pdf 文件。


开始
输入year、month、day
1
int[] MonthDays = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
2
((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0
3
F
T
MonthDays[1] = 29
4
year < 0 || month < 0 || day < 0
5
F
year > 10000
7
F
month > 12
9
F
day > MonthDays[month - 1]
11
F
totalDays = 0; i = 1
13
T
return 0
6
T
return -1
8
T
return -2
10
T
return -3
12
结束
i < year
14
T
F
((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0
15
T
F
totalDays += 365;
17
totalDays += 366;
16
i=i+1
18
i=1
19
i < month
20
T
F
i == 2
21
T
F
((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0
22
T
F
totalDays += MonthDays[i - 1]; i+=1
24
MonthDays[1] = 29
23
totalDays += day int tempDay = totalDays % 7
25
tempDay == 0
26
T
F
tempDay = 7
27
return tempDay
28

II. 语句覆盖和判定覆盖测试用例

编号	执行条件	输入 (year, month, day)	期望输出	实际输出	执行路径
01	语句覆盖	-1, -1, -1	0	0	1-2-3-5-6
02	语句覆盖	20000, 1, 1	-1	-1	1-2-3-4-5-7-8
03	语句覆盖	1, 14, 1	-2	-2	1-2-3-5-7-9-10
04	语句覆盖	400, 2, 30	-3	-3	1-2-3-4-5-7-9-11-12
05	语句覆盖	5, 1, 23	7	7	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-14-15-16-18-19-25-26-27-28
06	语句覆盖	4, 3, 2	2	2	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-19-20-21-24-20-21-22-23-24-25-26-28

编号	执行条件	输入 (year, month, day)	期望输出	实际输出	判定取值 (3, 5, 7, 9, 11, 14, 15, 20, 21, 22, 26)	执行路径
01	判定覆盖	-1, -1, -1	0	0	FT////////	1-2-3-5-6
02	判定覆盖	20000, 1, 1	-1	-1	TFT///////	1-2-3-4-5-7-8
03	判定覆盖	1, 14, 1	-2	-2	FFFT//////	1-2-3-5-7-9-10
04	判定覆盖	400, 2, 30	-3	-3	TFFFT/////	1-2-3-4-5-7-9-11-12
05	判定覆盖	5, 1, 23	7	7	FTTTT(TTTTF)(FFFT)F//T	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-14-15-16-18-19-25-26-28
06	语句覆盖	4, 3, 2	2	2	FTTTT(TTT)(FFF)(TTF)(FT)(FT)F	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-19-20-21-24-20-21-22-23-24-25-26-28

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：-1 -1 -1
输入日期为：星期0

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：20000 1 1
输入日期为：星期-1

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1 14 1
输入日期为：星期-2

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：400 2 30
输入日期为：星期-3
```

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：4 3 2
输入日期为：星期2

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：5 1 23
输入日期为：星期7
```

III. 条件覆盖、判定条件覆盖和条件组合覆盖测试用例

编号	执行条件	输入 (year, month, day)	期望输出	实际输出	条件取值 (3, 5, 15, 22)	执行路径
01	条件覆盖	-1, -1, -1	0	0	FTF T// /// ///	1-2-3-5-6
02	条件覆盖	1, -1, 9	0	0	FTF FT/ /// ///	1-2-3-5-6
03	条件覆盖	1, 1, -9	0	0	FTF FFT /// ///	1-2-3-5-6
04	条件覆盖	8, 3, 2	7	7	TT/ FFF TT/|FTF TT/	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-19-20-21-24-20-21-22-23-24-25-26-27-28
05	条件覆盖	2000, 2, 1	2	2	TFT FFF FFF|TFT|TT/|FTF|TFF TFT	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 1999 次 -19-20-21-24-20-25-26-28
06	条件覆盖	1999, 2, 1	1	1	FTF FFF FFF|TFT|TT/|FTF|TFF FTF	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 1998 次 -19-20-21-24-20-25-26-28

编号	执行条件	输入(year,month,day)	期望输出	实际输出	条件取值(3,5,15,22)	执行路径
01	判定条件覆盖	-1,-1,-1	0	0	F T / / FTF T// /// ///	1-2-3-5-6
02	判定条件覆盖	1,-1,9	0	0	F T / / FTF FT/ /// ///	1-2-3-5-6
03	判定条件覆盖	1,1,-9	0	0	F T / / FTF FFT /// ///	1-2-3-5-6
04	判定条件覆盖	8,3,2	7	7	T F T|F T TT/ FFF TT/|FTF TT/	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-19-20-21-24-20-21-22-23-24-25-26-27-28
05	判定条件覆盖	2000,2,1	2	2	F F F|T F TFT FFF FFF|TFT|TT/|FTF|TFF TFT	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 1999 次 -19-20-21-24-20-25-26-28
06	判定条件覆盖	1999,2,1	1	1	F F F|T F FTF FFF FFF|TFT|TT/|FTF|TFF FTF	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 1998 次 -19-20-21-24-20-25-26-28

编号	执行条件	输入 (year, month, day)	期望输出	实际输出	条件取值 (3, 5, 15, 22)	执行路径
01	条件组合覆盖	-1, -1, -1	0	0	FTF T// /// ///	1-2-3-5-6
02	条件组合覆盖	1, -1, 9	0	0	FTF FT/ /// ///	1-2-3-5-6
03	条件组合覆盖	1, 1, -9	0	0	FTF FFT /// ///	1-2-3-5-6
04	条件组合覆盖	8, 3, 2	7	7	TT/ FFF TT/|FTF TT/	1-2-3-5-7-9-11-13-14-15-17-18-14-15-17-18-14-15-17-18-19-20-21-24-20-21-22-23-24-25-26-27-28
05	条件组合覆盖	2000, 2, 1	2	2	TFT FFF FFF|TFT|TT/| FTF|TFF TFT	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 -19-20-21-24-20-25-26-28
06	条件组合覆盖	1999, 2, 1	1	1	FTF FFF FFF|TFT|TT/| FTF|TFF FTF	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 -19-20-21-24-20-25-26-28
07	条件组合覆盖	3000, 2, 1	6	6	FFF FFF FFF|TFT|TT/| FTF|TFF FFF	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 -19-20-21-24-20-25-26-28
08	条件组合覆盖	1400, 2, 1	6	6	TFF FFF FFF|TFT|TT/| FTF|TFF TFF	1-2-3-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 -19-20-21-24—20-25-26-28

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1999 2 1
输入日期为：星期1

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：3000 2 1
输入日期为：星期6

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1400 2 1
输入日期为：星期6

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1 -1 9
输入日期为：星期0
```

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1 1 -9
输入日期为：星期0

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：8 3 2
输入日期为：星期7

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：2000 2 1
输入日期为：星期2
```

IV.对循环的测试用例

编号	输入 (year,month,day)	期望输出	实际输出	执行路径
01	2000，1，30	7	7	1-2-3-4-5-7-9-11-13-14-15-17-18-14-15-16-18 循环 n 次 -19-20-25-26-27-28
02	1，12，20	4	4	1-2-3-5-7-9-11-13-14-19-20-21-24-20-循环 n 次 -25-26-28

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1 12 20
输入日期为：星期4

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：2000 1 30
输入日期为：星期7
```

V. 路径覆盖

如果按总共 10 个分支算，共有 1024 条路径，但因为 5，7，9，11 分支如果取 true 将直接返回，则共有 64+4*2=72 种，又由于有的分支是嵌套的且是循环的，因此共有 16+8+1+2=27 条。

包含不可达路径，因为语句 3 和 22 完全一样，不可能存在其中一个进入 T 另一个进入 F 的情况。

27 个用例我要设计三个小时。

VI. MC/DC 测试用例

For

```
…
int[] MonthDays = { 31, 28, 31, 30, 31, 30, 31,
31, 30, 31, 30, 31 };
if (((year % 4 == 0) && (year % 100 != 0)) ||
year % 400 == 0) {
        MonthDays[1] = 29;
}
…
```

编号	输入(year, month, day)	预期输出	实际输出	条件取值	判定取值
1	1, 12, 20	28 星期 4	28 星期 4	F T F	F
2	4, 1, 1	29 星期 4	29 星期 4	T T /	T
3	1400, 3, 7	28 星期 5	28 星期 5	T F F	F
4	2000, 1, 30	29 星期 7	29 星期 7	T F T	T

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1 12 20
28
输入日期为：星期4

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：4 1 1
29
输入日期为：星期4

C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：1400 3 7
28
输入日期为：星期5
```

```
C:\Users\zmj\Documents\Tencent Files\838818140\FileRecv>java WeekA
请输入日期（格式xxxx xx xx）：2000 1 3
29
输入日期为：星期1
```

FOR

```
...
if (((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0) {
        totalDays += 366;
    } else {
        totalDays += 365;
    }
...
```

编号	输入 i	输出 totalDays		条件取值	判定取值
		预期输出	实际输出		
1	1	365	365	F T F	F
2	4	366	366	T T /	T
3	100	365	365	T F F	F
4	400	366	366	T F T	T

```
i=1 totalDays+=365
i=2 totalDays+=365
i=3 totalDays+=365
i=4 totalDays+=366
```

```
i=100 totalDays+=365
```

```
i=400 totalDays+=366
```

VII. 条件短路对控制流测试的影响

源代码如下：

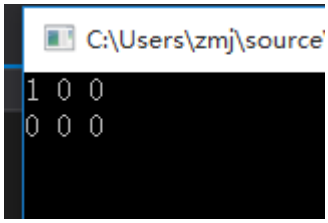
```
#include<iostream>
using namespace std;

void main()
{
    int x = 0, y = 0, z = 0;
    if ((x = 1) || (y = 1 )|| (z = 1))
    {
        cout<<x << " " << y << " " << z<<endl;;
    }
    x = y = z = 0;
    if ((x == 1) && (y = 1) && (z = 1)) {}

    cout << x << " " << y << " " << z << endl;;

    cin.get();
    }
```

如果不发生逻辑短路，结果应该是 1 1 1 \n 0 1 1
但因为发生了逻辑短路，输出结果如下



三、实验体会

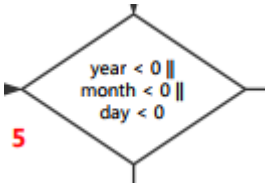
1. 对白盒测试的理解

通过这次实验，加深了对白盒测试的理解。

白盒测试是完全根据代码进行测试的一种测试，只能发现代码编写错误，而不

能发现程序运行是否满足需求。如本实验给的例子中，下图的判断应该考虑到年月不可能为零，但是，白盒测试并不能发现这一问题。在本次测试中，所有的预期结果和实际结果都相同，但这并不代表整个程序完全没有错误。不仅是因为逻辑错误无法被检查，还有一些重复冗余的语句和可能抛出的异常都没有被检测到。

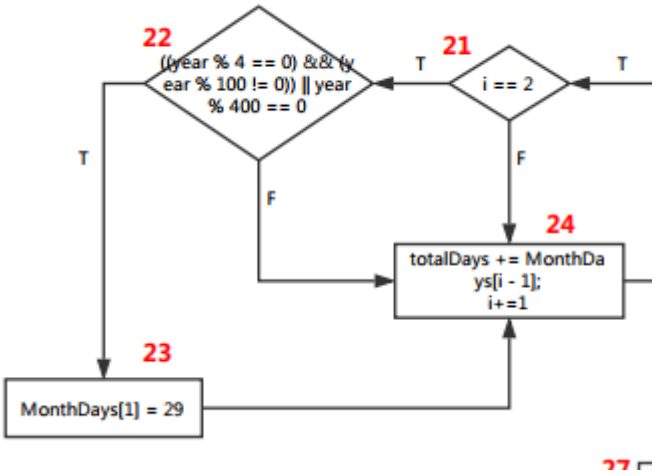
因此，白盒测试只能作为测试的一个环节，并不能作为测试的全部。

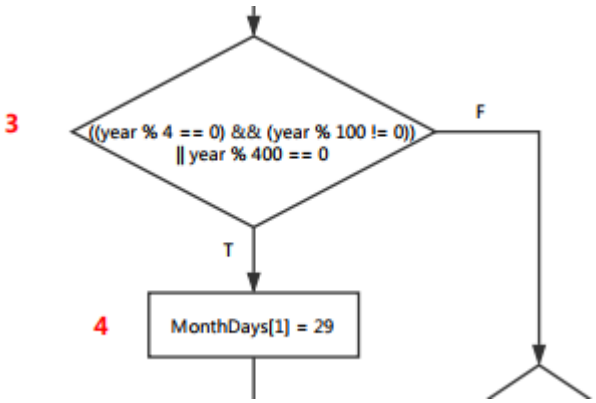


2. 程序中可能存在的错误

如下图所示，两处出现完全相同的语句，其作用也完全相同，完全可以删掉一处。

没有公元零年，month=0 数组越界，抛出异常的情况未予以考虑。





3. 对软件测试的理解

软件测试真的是一项很费精力的工作，尤其是用人工找出对应的用例，希望软件测试过程能有更好用的工具出现，减少测试人员的开销。