

7.1 编码

7.2 软件测试基础

7.6 白盒测试技术

7.7 黑盒测试技术

7.3 单元测试

7.4 集成测试

7.5 确认测试

7.8 调试

7.7 黑盒测试技术

□ **黑盒测试着重测试软件功能。并不能取代白盒测试，与白盒测试互补，主要用于测试过程的后期**

□ **设计黑盒测试方案时，应该考虑下述问题。**

- (1) 怎样测试功能的有效性？
- (2) 哪些类型的输入可构成好的测试用例？
- (3) 系统是否对特定的输入值特别敏感？
- (4) 怎样划定数据类的边界？
- (5) 系统能够承受什么样的数据率 and 数据量？
- (6) 数据的特定组合将对系统运行产生什么影响？

7.7 黑盒测试技术



7.7 黑盒测试技术

一、等价类划分

□ 根据数据测试的等效性原理进行划分。

数据测试的等效性：分类的数据取其子集中一个数据做测试与子集中其他数据测试的效果是等效的。

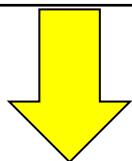
- 如果子集中的一个数据能测出软件错误，则子集中的其余数据也能测出错误；
- 相反，子集中一个数据测试不出程序错误，子集中的其余数据也测不出错误。

7.7 黑盒测试技术

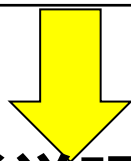
等价类：某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是等效的。

有效等价类：对于程序的规格说明来说，是合理的，有意义的输入数据构成的集合

无效等价类：对程序的规格说明来说，是不合理的，无意义的输入数据构成的集合



检验程序是否能实现
预期的功能和性能



找出程序异常说明情况，检查程序功能和性能的实现是否有不符合规格说明要求的地方

同时考虑

7.7 黑盒测试技术

□ 等价类划分方法：

①把程序的**输入数据集合**按输入条件划分为若干等价类，每个等价类相对于输入条件表示为一组有效或无效的输入，然后为每一等价类设计一个测试用例。

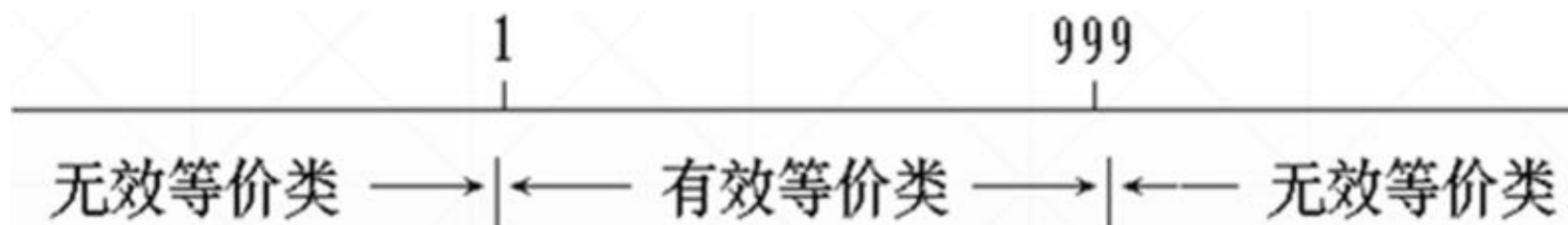
②确定输入等价类时还需分析**输出数据的等价类**，以便根据输出数据的等价类导出对应的输入等价类。

等价类的划分在很大程度上是一个探索性的过程，主要依靠测试人员的经验，下面几点仅供参考。

7.7 黑盒测试技术

(1) 如果规定了**输入值的范围**，可划分出一个有效的等价类(输入值在此范围内)，两个无效的等价类(输入值小于最小值或大于最大值)。

■ 输入值的范围：数值为1 ~ 999



(2) 如果规定了输入数据的个数，则类似地也可以划分出一个有效的等价类和两个无效的等价类。

■ **输入数据的个数： 每名学生一学期只能选1 ~ 3门课程**

□ **一个有效等价类： 选修1 ~ 3门课程**

□ **两个无效等价类： 不选修和选修超过3门**

(3) 如果规定了输入数据的一组值，而且程序对不同输入值做不同处理，则每个允许的输入值是一个有效的等价类，此外还有一个无效的等价类(任一个不允许的输入值)。

■ **可能的值：** 出差时交通工具的类型必须是火车、汽车或轮船，每种类型的报销额度不同。

□ **三个有效等价类：** 火车、汽车和轮船三种

□ **一个无效等价类：** 飞机

(4) 如果规定了输入数据必须遵循的规则**，则可划分出一个有效等价类(符合规则)和若干无效等价类(不同角度违反规则)**

■ **必须成立的条件：标识符的第一个字符必须是字母**

□ **一个有效等价类：第一个字符是字母**

□ **一个无效等价类：第一个字符不是字母**

(5) 如果输入条件是一个**布尔量**，则可以确定一个有效等价类和一个无效等价类。

(6) 如果规定了**输入数据为整型**，则可划分出正整数、零和负整数3个有效类。

(7) 如果程序的**处理对象是表格**，则应该使用空表，以及含一项或多项的表。

7.7 黑盒测试技术

□ 划分出等价类以后，根据等价类设计测试方案时主要使用下面两个步骤。

(1) 设计一个新的测试方案以尽可能多地覆盖尚未被覆盖的有效等价类，重复这一步骤直到所有有效等价类都被覆盖为止

(2) 设计一个新的测试方案，使它覆盖一个(且只覆盖一个)尚未被覆盖的无效等价类，重复这一步骤直到所有无效等价类都被覆盖为止。

注意：通常程序发现一类错误后就不再检查是否还有其他错误，因此，应该使每个测试方案只覆盖一个无效等价类。

7.7 黑盒测试技术

例：变量的命名规则一般规定如下：变量名的长度不能多于40个字符，第一个字符必须为英文字母，其他字母可以是英文字母、数字以及下划线的任意组合。请用等价分类法设计测试用例。

(1) 划分等价类：先考虑输入条件

输入条件	有效等价类	无效等价类
长度	(1) 小于 40 个字符 (2) 等于 40 个字符	(5) 大于 40 个字符
第一个字符	(3) 英文字母	(6) 非英文字母
其他字母	(4) 英文字母、数字或下划线的任意组合	(7) 空格 (8) 标点符号 (9) 运算符 (10) 其他可显示字符

7.7 黑盒测试技术

输入条件	有效等价类	无效等价类
长度	(1) 小于 40 个字符 (2) 等于 40 个字符	(5) 大于 40 个字符
第一个字符	(3) 英文字母	(6) 非英文字母
其他字母	(4) 英文字母、数字或下划线的任意组合	(7) 空格 (8) 标点符号 (9) 运算符 (10) 其他可显示字符

(2) 为合理等价类设计测试用例：

测试数据	测试范围	期望结果
(1) s_name12	等价类(1),(3),(4)	有效
(2) a1b2c3d4e5f6g7h8i9j1k2l3m4n5o6p7q8r9s_tr	等价类(2),(3),(4)	有效

7.7 黑盒测试技术

输入条件	有效等价类	无效等价类
长度	(1) 小于 40 个字符 (2) 等于 40 个字符	(5) 大于 40 个字符
第一个字符	(3) 英文字母	(6) 非英文字母
其他字母	(4) 英文字母、数字或下划线的任意组合	(7) 空格 (8) 标点符号 (9) 运算符号 (10) 其他可显示字符

(3)为每个不合理等价类至少设计一个测试用例：

测试数据	测试范围	期望结果
(3) a1b2c3d4e5f6g7h8i9j1k2l3m4n5o6p7q8r9s_trff	等价类(5)	无效
(4) 234name	等价类(6)	无效
(5) ab gh	等价类(7)	无效
(6) ab!2f	等价类(8)	无效
(7) fg+ghh	等价类(9)	无效
(8) H@gh	等价类(10)	无效

7.7 黑盒测试技术

□实例：用户注册页面需求

- USER NAME: 长度为 3-19 ; 以字母开头
- 姓名: 非空
- 密码: 非空
- 确认密码: 值和密码值相同
- 出生月份: 年——四位数字; 月——1-12; 日——1-31

合理范围

网易 163.com 中国第一大电子邮件服务商

欢迎注册网易免费邮！您可以选择注册163、126、yeah.net三大免费邮箱。

创建您的帐号

用户名: * 检测

密码: *

再次输入密码: *

安全信息设置 (以下信息非常重要, 请慎重填写)

密码保护问题: *

密码保护问题答案: *

性别: * ☐ 男 ☐ 女

出生日期: * 年 月 日

7.7 黑盒测试技术

输入条件	有效等价类	无效等价类
USER NAME	长 3-19 (1)	长度<3 (2) 长度>19 (3)
	以字母开头 (4)	非字母开头 (5)
姓名	非空 (6)	为空 (7)
密码	非空 (8)	为空 (9)
确认密码	值和密码值相同 (10)	值和密码值不同 (11)
出生年份 (年)	年——四位 (12)	不是四位 (13)
	年——数字 (14)	年数中有字母或其他非数字符号 (15)
	年——合理范围 (16)	年数在不合理范围 (17)
出生年份 (月)	1—12 (18)	月数<1 (19) 月数>12 (20) 月数中有字母或其他非数字符号 (21)
出生年份 (日)	1—31 (22)	日数<1 (23) 日数>31 (24) 日数中有字母或其他非数字符号 (25)

7.7 黑盒测试技术

覆盖用例	输入					预期结果
	USER NAME	姓名	密码	确认密码	出生年份	
1、4、6、8、10、12、14、16、18、22	A123	weind	1	1	2000-5-6	成功注册
2	A1	weind	1	1	2000-5-6	失败
3	A1234567890123456789	weind	1	1	2000-5-6	失败
5	1123	weind	1	1	2000-5-6	失败
7	A123		1	1	2000-5-6	失败
9	A123	weind		1	2000-5-6	失败
11	A123	weind	1	2	2000-5-6	失败
13	A123	weind	1	1	20000-5-6	失败

7.7 黑盒测试技术

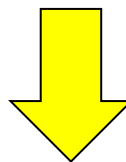
覆盖用例	输入					预期结果
	USER NAME	姓名	密码	确认密码	出生年份	
15	A123	weind	1	1	200a-5-6	失败
17	A123	weind	1	1	9999-5-6	失败
19	A123	weind	1	1	2000-0-6	失败
20	A123	weind	1	1	2000-17-6	失败
21	A123	weind	1	1	2000-a1-6	失败
23	A123	weind	1	1	2000-5-0	失败
24	A123	weind	1	2	2000-5-89	失败
25	A123	weind	1	1	2000-5-a1	失败

二、边界值分析

- 经验表明，处理边界情况时程序最容易发生错误。例如，许多程序错误出现在下标、纯量、数据结构和循环等等的边界附近。因此，设计使程序运行在边界情况附近的测试方案，暴露出程序错误的可能性更大一些。
- 使用边界值分析方法设计测试方案**首先应该确定边界情况**，通常输入等价类和输出等价类的边界。选取的测试数据应该**刚好等于、刚刚小于和刚刚大于边界值**。
- 通常设计测试方案时总是联合使用等价划分和边界值分析两种技术。

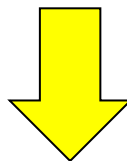
7.7 黑盒测试技术

例：三角形计算时，要输入三角形的三个边长：A、B和C。这三个数值应当满足 $A > 0$ 、 $B > 0$ 、 $C > 0$ 、 $A + B > C$ 、 $A + C > B$ 、 $B + C > A$ ，才能构成三角形。



边界情况

不等式中的任何一个“ $>$ ”错写成“ \geq ”，就不能构成三角形。



选取正好等于，刚刚大于，或刚刚小于边界的值做为测试数据

A、B、C比0稍大、等于0，任意两边和比第三条边稍大、等于

7.7 黑盒测试技术

三角形问题的边界值测试用例

序号	输入条件			预期输出
	A	B	C	
1	100	100	1	等腰三角形
2	100	100	0	非三角形
3	100	1	100	等腰三角形
4	100	0	100	非三角形
5	1	100	100	等腰三角形
6	0	100	100	非三角形
7	100	100	199	等腰三角形
8	100	100	200	非三角形
9	101	199	99	三角形
10	100	200	100	非三角形
11	199	100	100	等腰三角形
12	200	100	100	非三角形

三、错误推测

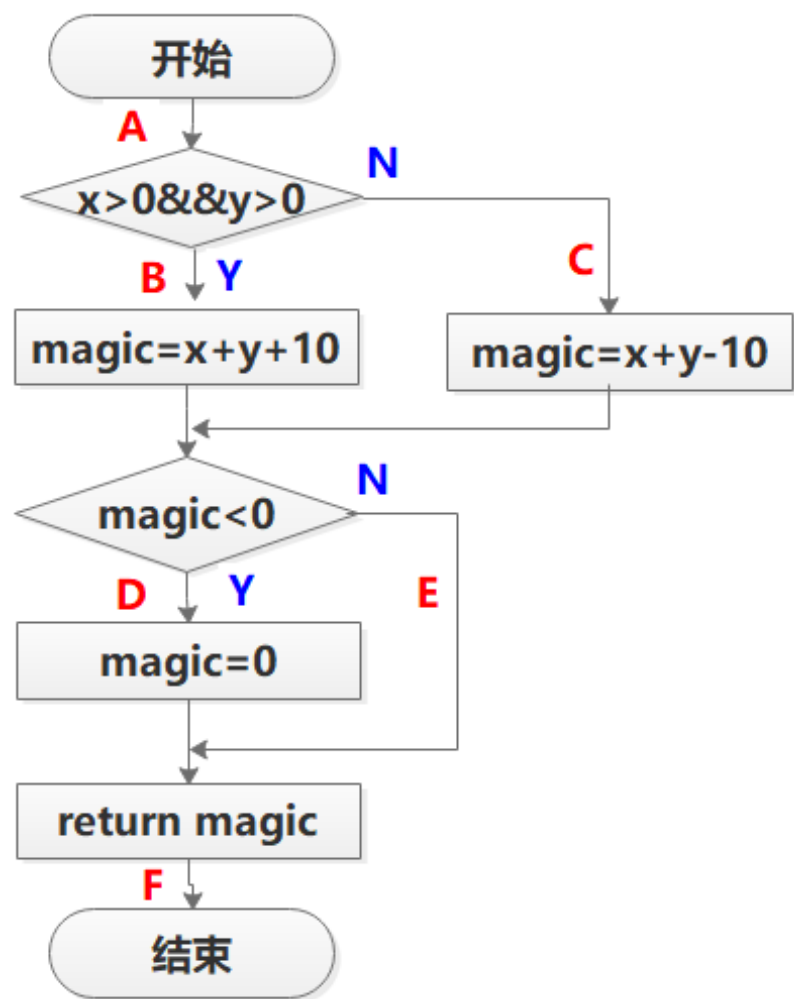
- ❑ **错误推测法**在很大程度上靠直觉和经验进行。它的基本想法是列举出程序中可能有的错误和容易发生错误的特殊情况，并且根据它们选择测试方案。
- ❑ 应该仔细分析程序规格说明书，找出其中遗漏或省略的部分，设计相应的测试方案检测对这些部分的处理是否正确
- ❑ 经验表明，在一段程序中已经发现的错误数目往往和尚未发现的错误数成正比。因此，在进一步测试时要着重测试那些已发现了较多错误的程序段。

7.7 黑盒测试技术

例：测试一个对线性表（比如数组）进行排序的程序，可推测列出以下几项需要特别测试的情况：

- 1、输入的线性表为空表；**
 - 2、表中只含有一个元素；**
 - 3、输入表中所有元素已排好序；**
 - 4、输入表已按逆序排好；**
 - 5、输入表中部分或全部元素相同**
-**

- ❑ 等价划分法和边界值分析法都只孤立地考虑各个输入数据的测试功效，而没有考虑多个输入数据的组合效应，可能会遗漏了输入数据易于出错的组合情况。
- ❑ 选择输入组合的一个有效途径是利用判定表或判定树为工具，列出输入数据各种组合与程序应作的动作(及相应的输出结果)之间的对应关系，然后为判定表的每一列至少设计一个测试用例。

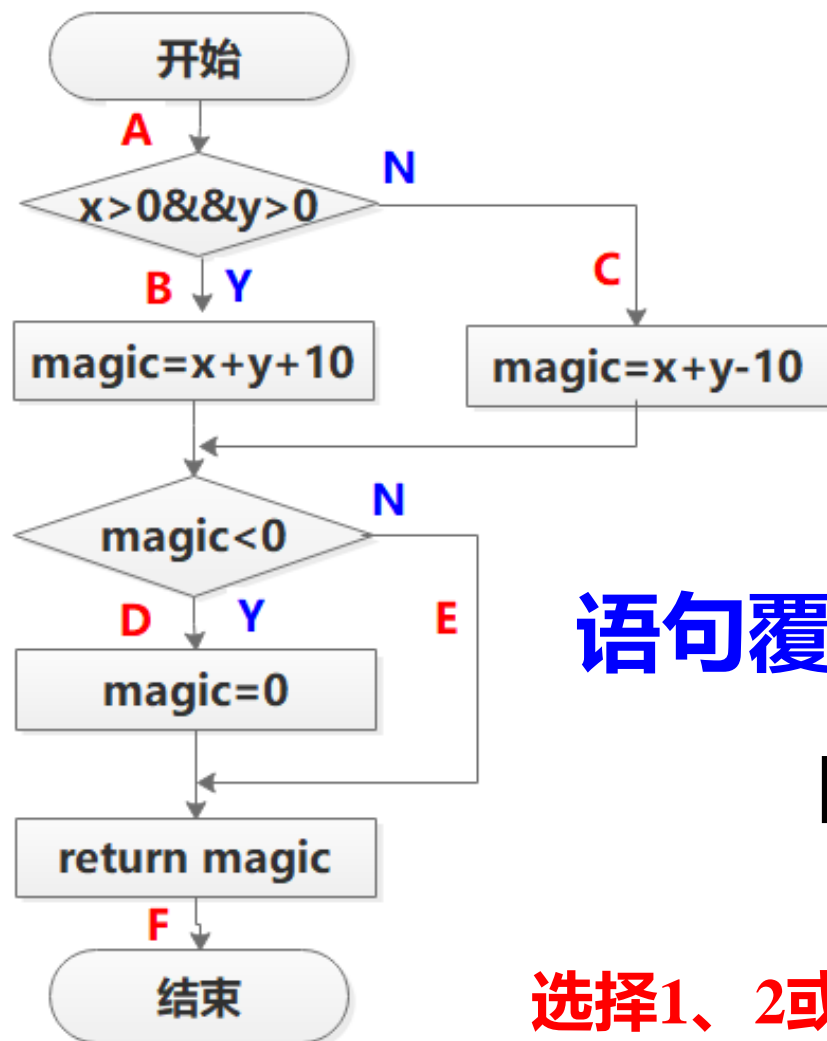


给出满足语句覆盖[填空1]和路径覆盖 [填空2] 的最少测试用例。
可供选择的测试数据见下表。

	x	y
测试数据组①	3	3
测试数据组②	3	0
测试数据组③	-3	3
测试数据组④	-3	0
测试数据组⑤	12	0

作答

序号	测试用例		条件取值			判定		路径	覆盖标准
	x	y	a1	a2	b	a	b		



	x	y
测试数据组①	3	3
测试数据组②	3	0
测试数据组③	-3	3
测试数据组④	-3	0
测试数据组⑤	12	0

语句覆盖

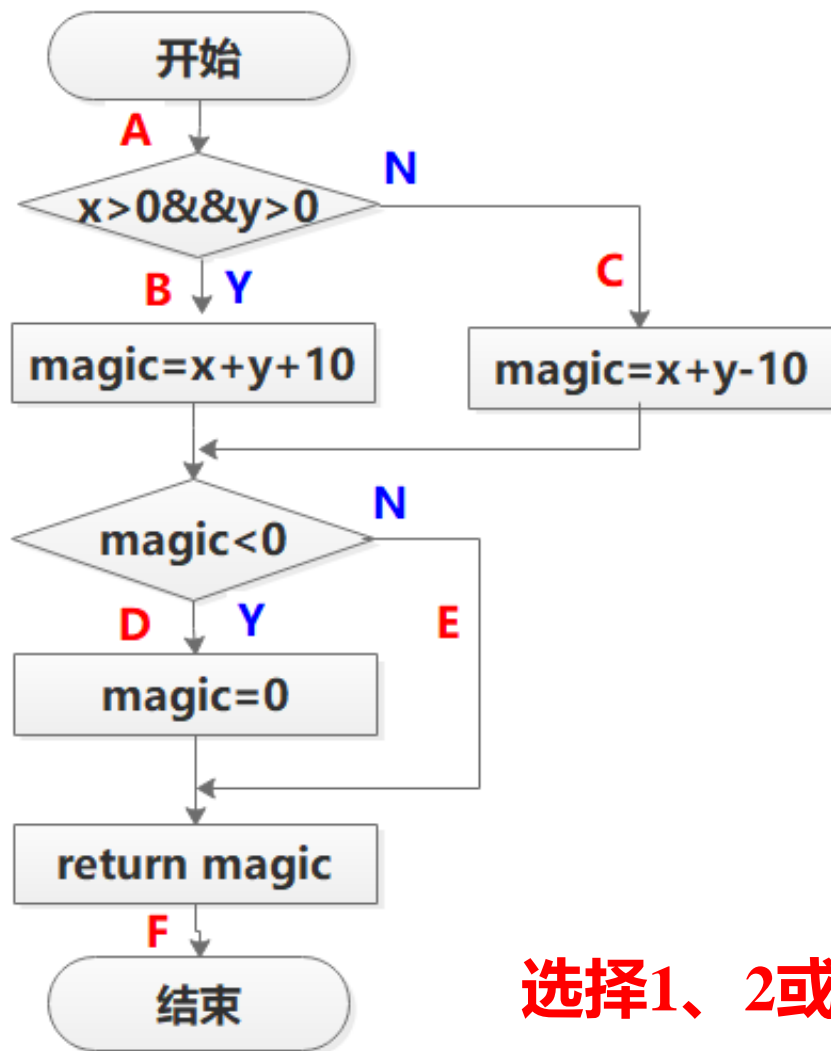
四条路径

A DF
ABEF
ACDF
ACEF

选择1、2或者1、3作为测试用例也可

序号	测试用例		条件取值			判定		路径	覆盖标准
	x	y	a1	a2	b	a	b		
1	3	3				1	0	ABEF	语句覆盖
2	-3	0				0	1	ACDF	

判定覆盖



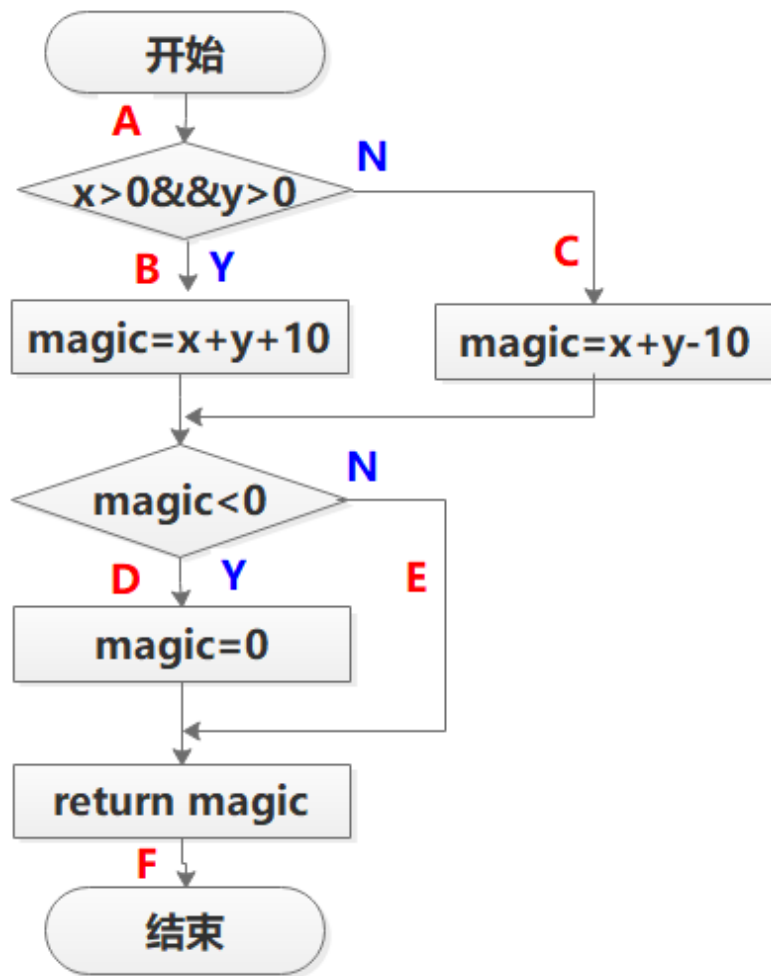
	x	y
测试数据组①	3	3
测试数据组②	3	0
测试数据组③	-3	3
测试数据组④	-3	0
测试数据组⑤	12	0

选择1、2或者1、3作为测试用例也可

序号	测试用例		条件取值			判定		路径	覆盖标准
	x	y	a1	a2	b	a	b		
1	3	3				1	0	ABEF	判定覆盖
2	-3	0				0	1	ACDF	

条件覆盖

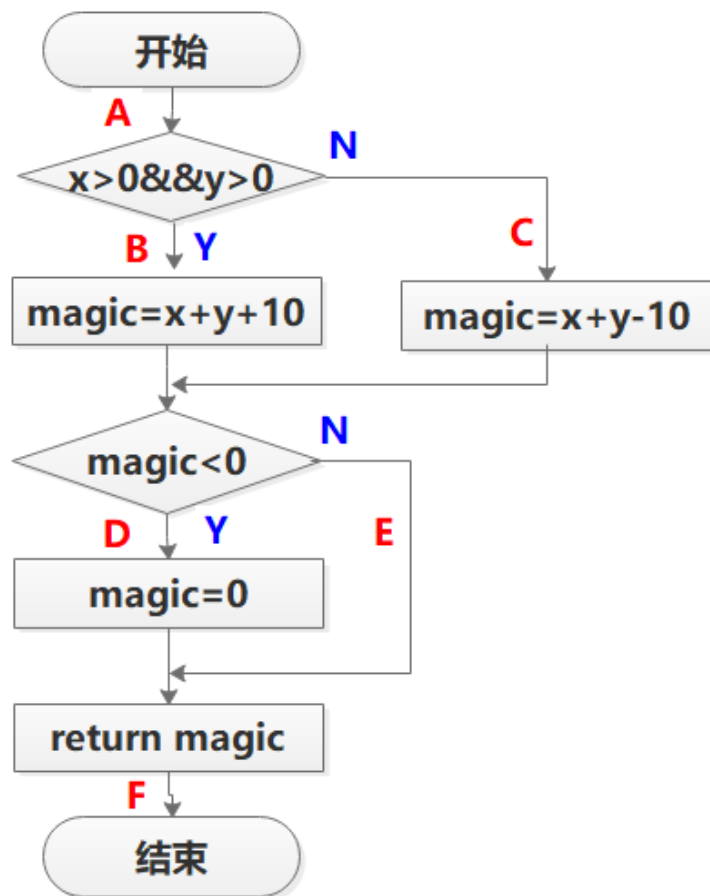
判定/条件覆盖



	x	y
测试数据组①	3	3
测试数据组②	3	0
测试数据组③	-3	3
测试数据组④	-3	0
测试数据组⑤	12	0

序号	测试用例		条件取值			判定		路径	覆盖标准
	x	y	a1	a2	b	a	b		
1	3	3	1	1	0	1	0	ABEF	条件覆盖 (判定/条件)
2	-3	0	0	0	1	0	1	ACDF	

条件组合覆盖



	x	y
测试数据组①	3	3
测试数据组②	3	0
测试数据组③	-3	3
测试数据组④	-3	0
测试数据组⑤	12	0

序号	测试用例		条件取值			判定		路径	覆盖标准
	x	y	a1	a2	b	a	b		
1	3	3	1	1	0	1	0	ABEF	条件组合覆盖
2	3	0	1	0	1	0	1	ACDF	
3	-3	3	0	1	1	0	1	ACDF	
4	-3	0	0	0	1	0	1	ACDF	



	x	y
测试数据组①	3	3
测试数据组②	3	0
测试数据组③	-3	3
测试数据组④	-3	0
测试数据组⑤	12	0

AXDF
ABEF
ACDF
ACEF

选择1、2、5或者1、4、5
作为测试用例也可

序号	测试用例		条件取值			判定		路径	覆盖标准
	x	y	a1	a2	b	a	b		
1	3	3	1	1	0	1	0	ABEF	路径覆盖
2	-3	3	0	1	1	0	1	ACDF	
3	×	×				1	1	ABDF	
4	12	0	1	0	0	0	0	ACEF	