1. 使用逻辑覆盖测试方法测试以下程序段。

void do (int x, int a, int b){

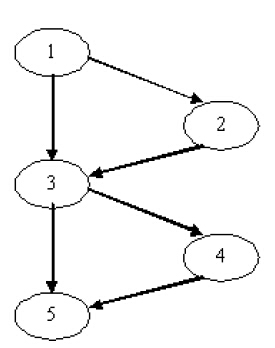
① if((a<4)&&(b==5)) ② x=x/4;

③ if((a==2)||(x>1)) ④x=x+1; ⑤ return ;}

(1)画出程序的控制流图。

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

(1)画出程序的控制流图。



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



1. 为以下程序段设计一组测试用例，要求分别满足语句覆盖、判定覆盖、条件覆盖。

void DoWork(int x, int y, int z){

int k=0, j=0; if((x>3)&&(z<10)) { k=x\*y-1; j=sqrt(k);}

if((x==4)||(y>5)) j=x\*y+10; j=j%3; }

答案：语句覆盖测试用例：x=4、y=5、z=5；

判定覆盖测试用例：x=4、y=5、z=5； x=2、y=5、z=5；

条件覆盖测试用例：x=4、y=6、z=5 ；x=2、y=5、 z=15 ；

1. 看代码程序：

void Sort(int m, int n) {

① int x=0; int y=0; ② while(m>0) {

③ if(iType==0) ④ x=y+2; ⑤ else if(n==1) ⑥ x=y+10;

⑦ else ⑧ x=y+20; ⑨}

⑩ Return ;}

画出控制流图；

计算圈复杂度并写出独立路径。

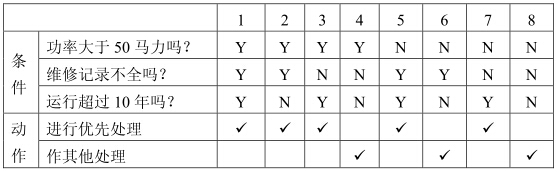
1. 某程序规定：“…对功率大于50马力的机器、维修记录不全或已运行10年以上的机器，应给予优先的维修处理…”。这里假定：“维修记录不全”和“优先维修处理”均已在别处有更严格的定义。请根据题目建立决策表。

答案：根据问题中描述的输入条件和输出结果，列出所有的条件桩和动作桩。其中

条件桩有三项：1）功率大于 50 马力吗？2）维修记录不全吗？3）运行超过 10 年吗？

动作桩有两项：1）进行优先处理；2）作其他处理；

判定表如下表所示：



1. 某个软件的规格说明中包含了下面的要求：第一列字符必须是A或B，第二列字符必须是一个数字，在此情况下进行文字的修改。但如果第一列字符不正确，则给出信息L，如果第二列字符非数字，则给出信息M。请用基于决策表的测试方法进行测试，并设计测试用例覆盖每一种情况。

答案：输入条件：第一列字符：{A}，{B}，{其他}；第二列字符：{数字}，{其他}

动作：修改文件，给出 L，给出 M。

决策表如下表所示：



1. 某公司招聘人员，其要求为：

学历：本科及以上；专业：计算机、通信、自动化； 年龄：22-30岁。

请划分出各条件的有效等价类和无效等价类。

（1）学历的等价类：有效等价类：A1={本科，硕士，博士}，

无效等价类：A2={专科，高中，初中，小学，无}

（2）专业的等价类：有效等价类：B1={计算机、通信、自动化}，无效等价类：

B2={其他}

（3）年龄的等价类：有效等价类：C1={22-30 岁}，

无效等价类：C2={小于 22 岁}，C3={大于 30 岁}

1. 假设中国某航空公司规定：

中国去欧美的航线所有座位都有食物供应。每个座位都可以播放电影。

中国去非欧美的国外航线都有食物供应，只有商务仓可以播放电影。

中国国内的航班的商务仓有食物供应，但是不可以播放电影。

中国国内的航班的经济仓仅在飞行时间大于2小时的情况下才有食物供应，但是不可以播放电影。

使用判定表法设计测试用例。

1. 使用逻辑覆盖测试方法测试以下程序段。

void do (int x, int a, int b)

{

if((a<4)&&(b=5))

x=x/4;

if((a=2)||(x>1))

x=x+1;

}

1. 画出程序的控制流图。
2. 分别以语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖和路径覆盖方法设计测试用例，并写出每个测试用例的执行路径。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 第二列 | 数字 | 数字 | 数字 | 其他 | 其他 | 其他 |
| 动作 | 修改文件 | √ | √ |  |  |  |  |
|  | 给出L |  |  | √ |  |  | √ |
|  | 给出M |  |  |  | √ | √ | √ |
| 测试用例 |  | A6 | B2 | M1 | A！ | B% | V+ |

9. 某公司人事软件的工资计算模块的需求规格说明书中描述：

（1）年薪制员工：严重过失，扣当月薪资的4%；过失，扣年终奖的2%；

（2）非年薪制员工：严重过失，扣当月薪资的8%；过失，扣当月薪资的4%。

根据题目内容列出条件和结果，给出决策表。

条件：c1：年薪制：c2：严重过失

结果：e1：扣月4% e2：扣月8% e3：扣月2%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 |
| 条件 | C1 | 1 | 1 | 0 | 0 |
|  | C2 | 1 | 0 | 1 | 0 |
| 动作 | E1 | √ |  |  | √ |
|  | E2 |  |  | √ |  |
|  | E3 |  | √ |  |  |

10. 请将以下程序分别按照语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖设计测试用例。

if((a>2&&b<3)&&(c>4||d<5))

{

statement1;

}

else

{

statement2;

}

11. 某一种8位计算机，其十六进制常数的定义是以0x或0X开头的十六进制整数，其取值范围为-7f～7f（不区分大小写字母），如0x13、0x6A、-0x3c。请采用等价类划分的方法设计测试用例。

12. 根据给出的程序代码，按要求回答问题。

void sort(int Num, int Type)

{

int x=0;

int y=0;

else{

if(Type==1)

x=y+5;

else

x=y+10;

}

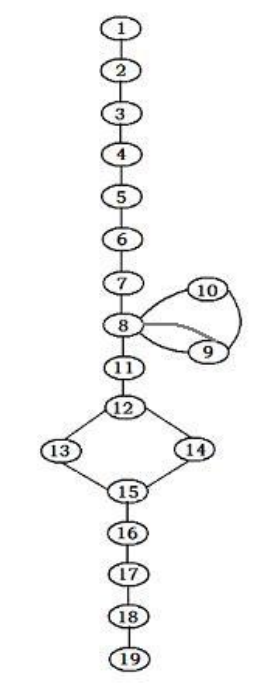
Num--;

}

}

1. 根据源程序画出控制流图；
2. 用三种方法计算上述控制流图的圈复杂度V(G)。
3. 找出独立路径。

13. 有一个处理单价为5角钱的饮料的自动售货机，其规格说明如下：若投入5角钱或1元钱的硬币，押下“橙汁”或“啤酒”的按钮，则相应的饮料就会送出来。若售货机没有零钱找，则一个显示“零钱找完”的红灯亮，这时再投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则显示“零钱找完”的红灯灭，在送出饮料的同时退还5角硬币。试根据规格说明书建立判定表。

****14. 根据给出的某程序流程图：

(1)计算它的环路复杂性；

(2)为了完成基本路径测试，求它的一组独立的路径。

15. 以试卷评分和成绩统计的程序为例，说明采用边界值分析法设计测试用例的过程。试卷评分和成绩统计的程序规格说明如下：

程序的输入条件由80个字符的记录组成。所有这些记录分为3组：

（1）标题；

这一组只有一个记录，其内容为输出报告的名字

（2）各题的标准答案记录（1-999道题）；

试卷各题标准答案记录：每个记录均在第80个字符处标以数字“2”。该组的第一个记录的第1至第3个字符为题目编号（取值1~999）。第10至59个字符给出第1至50题答案（每个合法字符表示一个答案）。该组的第2，第3，等等纪录相应为第51至第100，第101，第150，等等题答案。

（3）学生的答卷描述

该组每个记录的第80个字符均为数字“3“.每个学生的答卷在若干个记录中给出。

16. 某一种8位计算机，其十六进制常数的定义是以0x或0X开头的十六整数，其取值范围为-7f～7f



用例1：0x7F，覆盖等价类（1）（4）（6）（8）

用例2：-0Xb，覆盖等价类（1）（4）（6）（8）

用例3：0X0，覆盖等价类（1）（4）（6）（8）

用例4：0x，覆盖等价类（1）（7）

用例5：A7，覆盖等价类（2）

用例6：-1A，覆盖等价类（3）

用例7：0X8h，覆盖等价类（1）（5）

用例8：0x80，覆盖等价类（1）（4）（10）

用例9：-0XaB，覆盖等价类（1）（4）（9）

17. 输入：用户密码

要求：1）用户密码为6到8位；2）必须含有字母和数的组合。



18. 对招干考试系统“输入学生成绩”子模块设计测试用例。招干考试分三个专业,准考证号第一位为专业代号，如：①-行政专业，②-法律专业，③-财经专业。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入条件** | **有效等价类** | **编号** | **无效等价类** | **编号** |
| **行政专业准考证号码** | **110001～111215** | **（1）** | **负无穷～110000** | **（4）** |
| **法律专业准考证号码** | **210001～212006** | **（2）** | **111216～310000** | **（5）** |
| **财经专业准考证号码** | **310001～314015** | **（3）** | **314016～正无穷** | **（6）** |



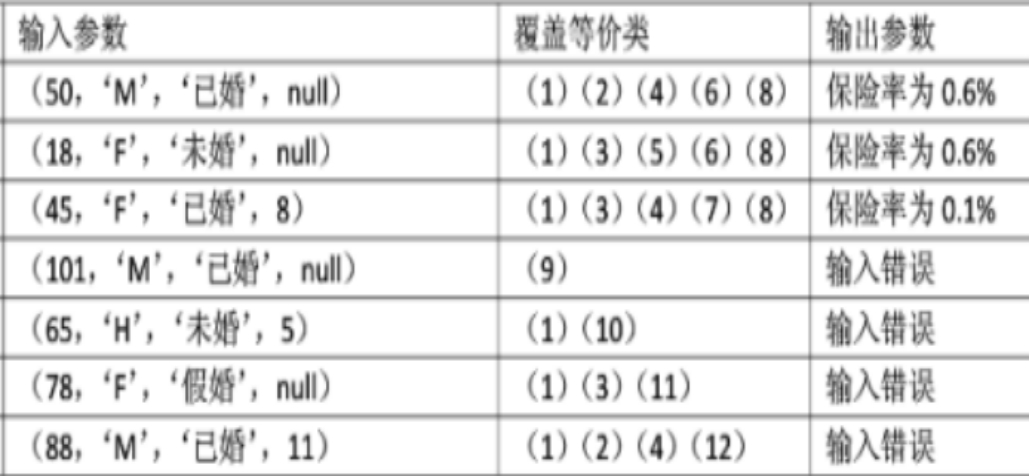
19.某城市的电话号码由3部分组成，这三部分的名称和内容分别是：地区码：空白或三位数字；前 缀：非‘0’或‘1’开头的三位数；后 缀：四位数字。

表格

描述已自动生成

20. 某保险公司的人寿保险的保费计算方式为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入条件** | **有效等价类** | **编号** | **无效等价类** | **编号** |
| **年龄** | **1~99整数** | **1** | **非1~99整数** | **9** |
| **性别** | **‘M’** | **2** | **非‘M’and非‘F’** | **10** |
| **‘F’** | **3** |
| **婚姻** | **‘已婚’** | **4** | **非‘已婚’and非‘未婚’** | **11** |
| **‘未婚’** | **5** |
| **扶养人数** | **null** | **6** | **非‘null’and非‘1-9整数’** | **12** |
| **1-9整数** | **7** |
| **点数** | **1-99整数** | **8** | **非1-99整数** | **13** |

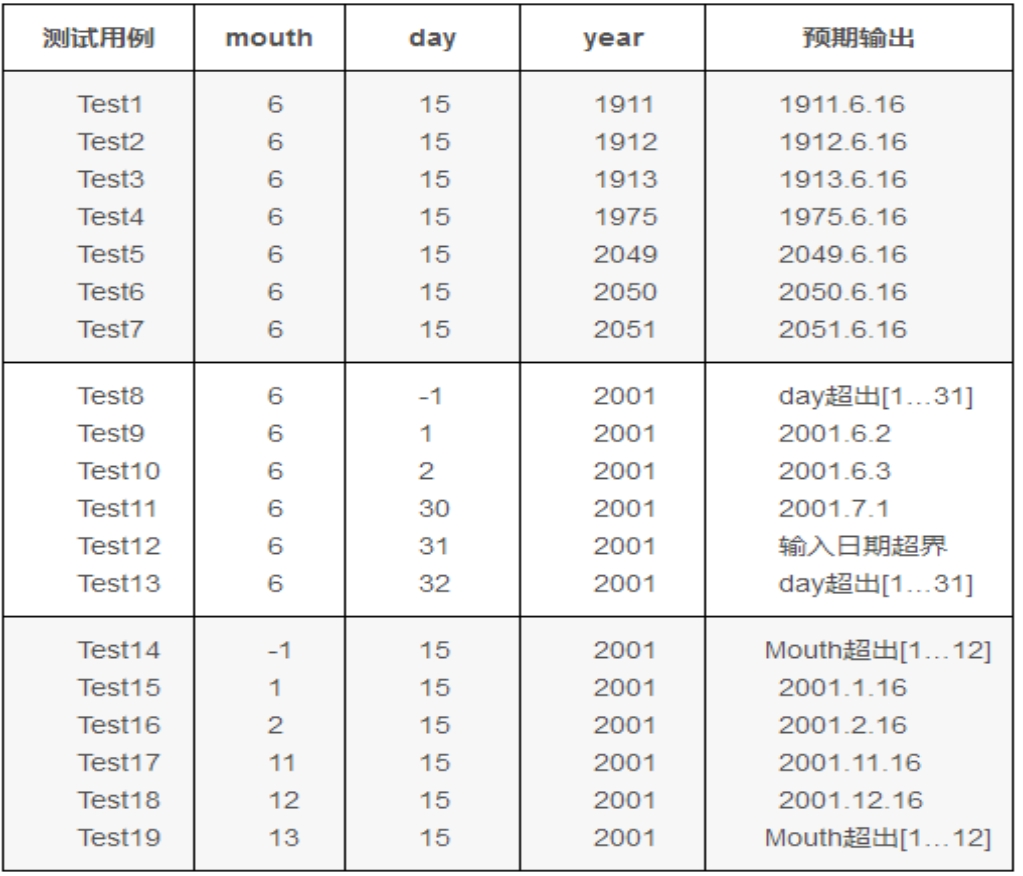


21. 输入三个整数a、b、c分别作为三边的边长构成三角形。

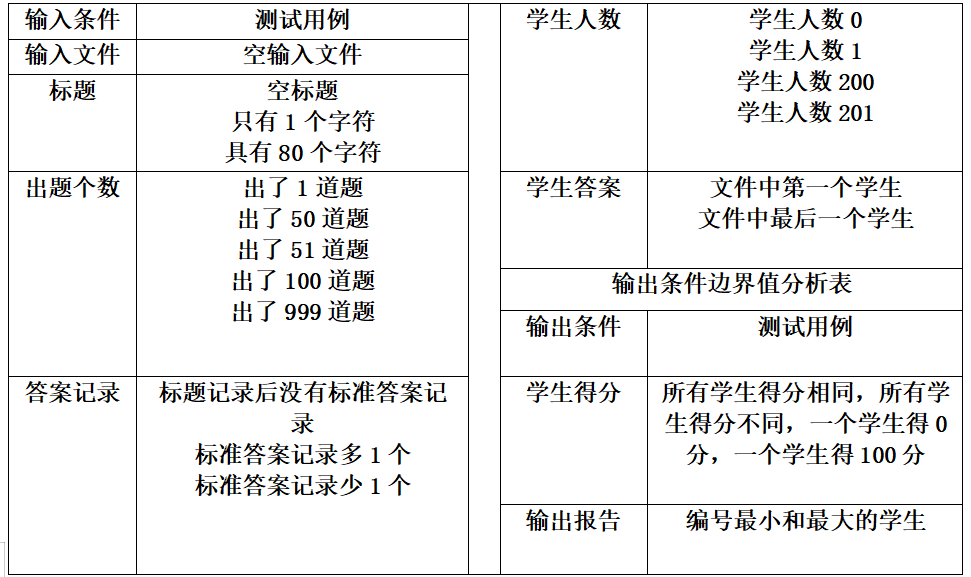
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入条件** | **有效等价类** | **编号** | **无效等价类** | **编号** |
| **是否是三角形两边** | **A>0** | **1** | **A<=0** | **7** |
| **B>0** | **2** | **B<=0** | **8** |
| **C>0** | **3** | **C<=0** | **9** |
| **A+B>C** | **4** | **A+B<=C** | **10** |
| **A+C>B** | **5** | **A+C<=B** | **11** |
| **B+C>A** | **6** | **B+C<=A** | **12** |
| **是否是等腰三角形** | **A=B** | **13** | **A≠B AND B≠C AND A≠C** | **16** |
| **B=C** | **14** |
| **A=C** | **15** |
| **是否是等边三角形** | **A=B AND B=C AND A=C** | **17** | **A≠B** | **18** |
| **B≠C** | **19** |
| **A≠C** | **20** |

测试用例没有固定答案，覆盖全部即可

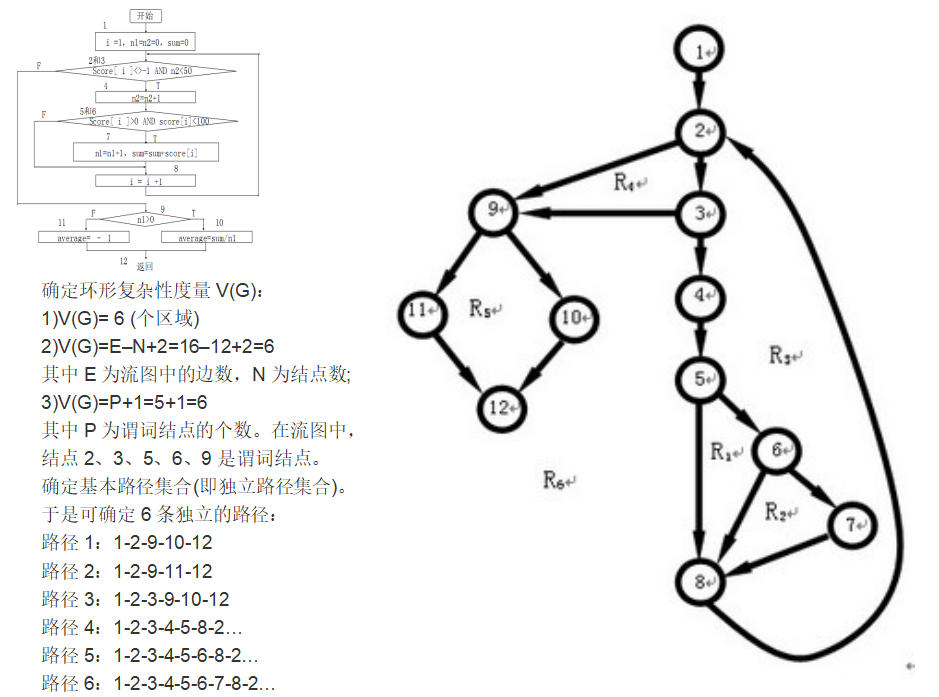
22. 为NextDate函数设计边界值分析测试用例。1≤month≤12和1≤day≤31，并设定变量year的取值范围为1912≤year≤2050。



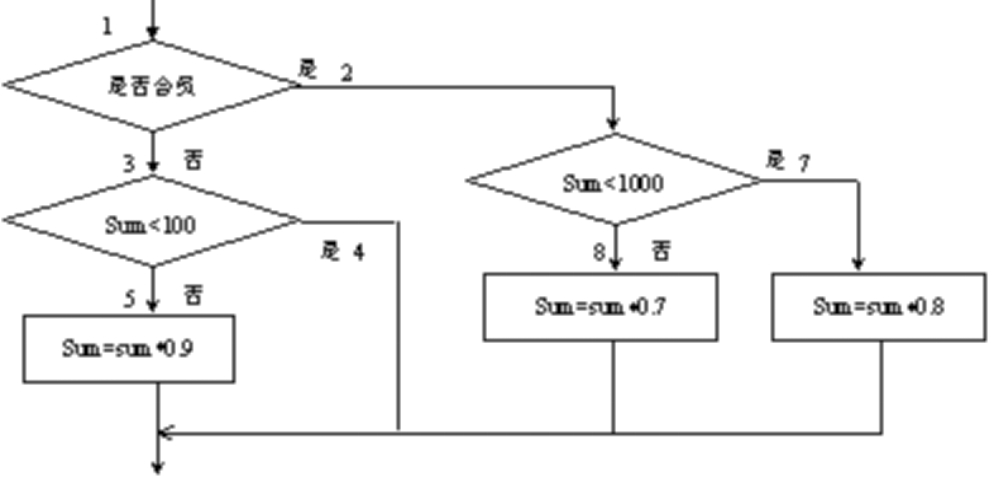
23. 以试卷评分和成绩统计的程序为例，说明采用边界值分析法设计测试用例的过程。



24. 下列程序流程图描述了最多输入50个值（以-1作为输入结束标志），计算其中有效的学生分数（0—100）的个数、总分数和平均值。做出控制流图，计算环形复杂度，给出程序的基本路径集（不需要给出测试用例）。



25. 某商场在“五一”期间，顾客购物时收费有4种情况



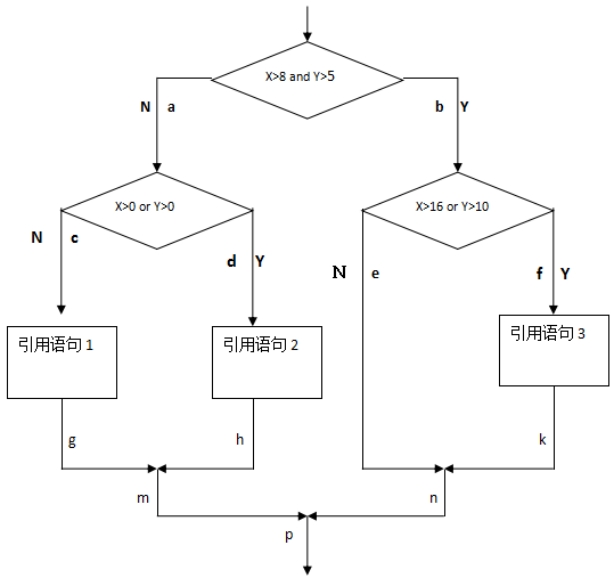
是会员，累计消费900元，覆盖路径1，2，7

是会员，累计消费2000元，覆盖路径1，2，8

不是会员，累计消费80元，覆盖路径1，3，4

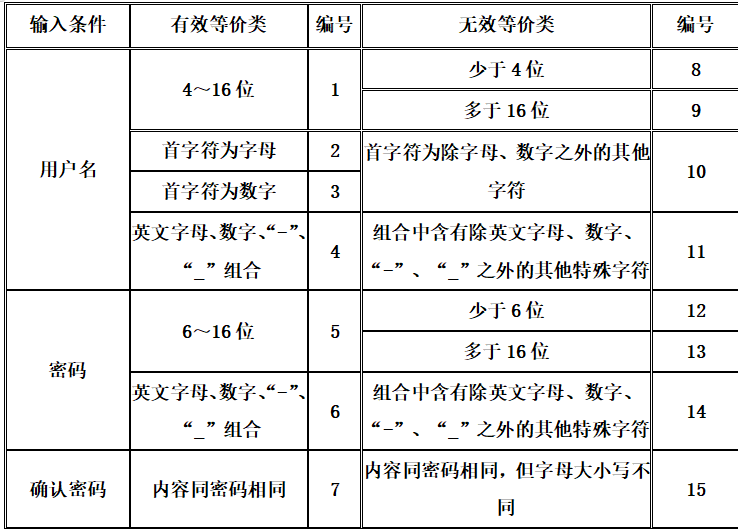
不是会员，累计消费300元，覆盖路径1，3，5

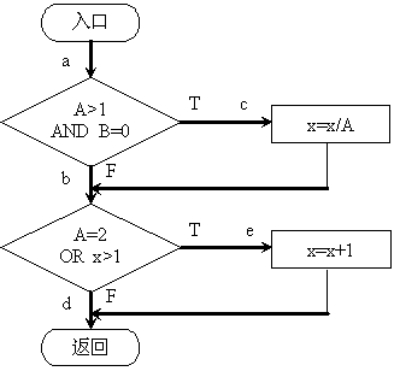
26. 为以下流程图所示的程序段设计一组测试用例，要求分别满足语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、组合覆盖和路径覆盖。



|  |  |  |
| --- | --- | --- |
| 语句覆盖 | 测试用例 | 覆盖语句 |
|  | X=0,Y=0 | 引用语句1 |
|  | X=1,Y=0 | 引用语句2 |
|  | X=17,Y=6 | 引用语句3 |
| 判定覆盖 | 测试用例 | 覆盖判定 |
|  | X=0,Y=0 | 判定1的N,判定2的N |
|  | X=1,Y=0 | 判定1的N,判定2的Y |
|  | X=15,Y=9 | 判定1的Y,判定2的N |
|  | X=17,Y=6 | 判定1的Y,判定2的Y |
| 将X>0，真为T3，假为F3  将Y>0，真为T4，假为F4  将X>16，真为T5，假为F5  将Y>16，真为T6，假为F6 | | |
| 条件覆盖 | 测试用例 | 覆盖条件 |
|  | X=0,Y=0 | F1.F2.F3.F4.F5.F6 |
|  | X=17,Y=11 | T1.T2.T3.T4.T5.T6 |
| 判定/条件覆盖 | | |
| 测试用例 | 覆盖条件 | 覆盖判定 |
| X=0,Y=0 | F1.F2.F3.F4 | 判定1的N,判定2的N |
| X=1,Y=1 | F1.F2.T3.T4 | 判定1的N,判定2的Y |
| X=17,Y=9 | T1.T2.F5.F6 | 判定1的Y,判定2的N |
| X=17,Y=11 | T1.T2.T5.T6 | 判定1的Y,判定2的Y |
| 条件组合覆盖  组合1：T1.T2 组合5：T3.T4 组合9：T5.T6  组合2：T1.F2 组合6：T3.F4 组合10：T5.F6  组合3：F1.T2 组合7：F3.T4 组合11：F5.T6  组合4：F1.F2 组合8：F3.F4 组合12：F5.F6 | | |
| 条件组合覆盖 | 测试用例 | 覆盖条件组合 |
|  | X=17,Y=11 | 1.5.9 |
|  | X=17,Y=0 | 2.6.10 |
|  | X=0,Y=11 | 3.7.11 |
|  | X=0,Y=0 | 4.8.12 |
| 路径覆盖  所有路径：a b c d e f g h k m n p | | |
| 路径覆盖 | 测试用例 | 覆盖路径 |
|  | X=0,Y=0 | a→c→g→m→p |
|  | X=1,Y=1 | a→d→h→m→p |
|  | X=17,Y=9 | b→e→n→p |
|  | X=17,Y=11 | b→f→k→n→p |

27. 在某网站申请免费信箱时，要求用户必须输入用户名、密码及确认密码，对每一项输入条件的要求如下：



28. 根据下图给出：

（1）语句覆盖测试用例

（2）判定覆盖测试用例

参考答案：

图中共有4条路径：P1（ace）、P2（abd）、

P3（abe）、P4（acd）。

（1）P1正好满足语句覆盖的条件。可以设计

如下的输入数据：

A＝2，B＝0，x＝4

（2）测试用例如果能够测试路径P1（ace）和P2（abd），就可以满足判定覆盖要求。可以设计如下两组输入数据：

A＝2，B＝0，x＝4

A＝1，B＝1，x＝1

29. 请将以下程序分别按照语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖设计测试用例。

**if((a>2&&b<3)&&(c>4||d<5)){statement1;}else{ statement2;}**

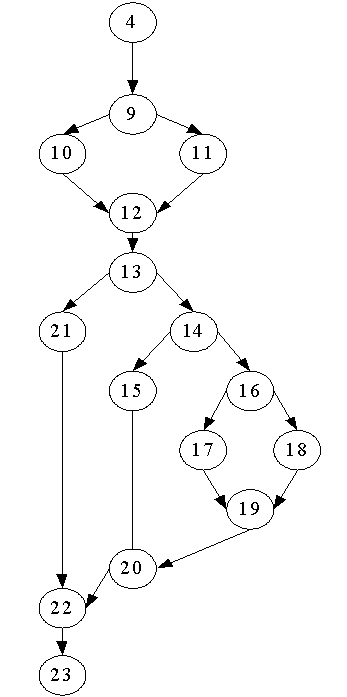
语句覆盖：(a=0.b=0.c=0.d=0)和(a=3.b=1.c=5.d=0)

判定覆盖：(a=0.b=0.c=0.d=0)和(a=3.b=1.c=5.d=0)

条件覆盖：(a=5.b=0.c=5.d=0)和(a=0.b=5.c=0.d=5)

判定/条件覆盖：(a=5.b=0.c=5.d=0)和(a=0.b=5.c=0.d=5)

30. 在三角形计算中，要求输入三角形的三个边长：A、B和C。当三边不可能构成三角形时，提示错误；可构成三角形时，计则打印给出程序伪代码、控制流程图、找出基本测试路径。

1. Program triangle2
2. Dim a,b,c As Integer
3. Dim IsATriangle As Boolean
4. Output(“Enter 3 integers which are sides of a triangle”)
5. Input(a,b,c)
6. Output(“Side A is ”,a)
7. Output(“Side B is ”,b)
8. Output(“Side C is ”,c)
9. If (a<b+c) AND(b<a+c)AND(c<a+b)
10. Then IsATriangle =True
11. Else IsATriangle =False
12. EndIf
13. If IsATrangle
14. Then if(a=b)AND (b=c)
15. Then Output(“Equilateral”)
16. Else If(a<>b)AND(a<>c)AND(b<>c)
17. Then Output(“Scalence”)
18. Else Output(“Isosecles”)
19. EndIf
20. EndIf
21. Else Output(“NOT a Triangle”)
22. EndIf
23. End triangle2

5条基本路径：

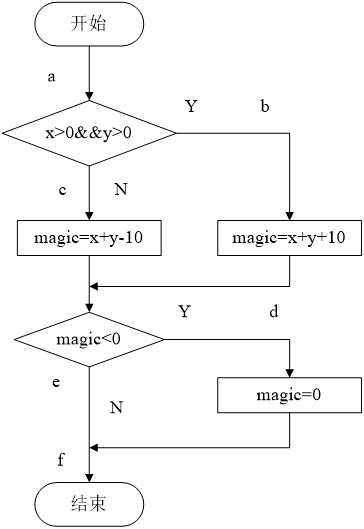
A.4-9-10-12-13-21-22-23

B.4-9-11-12-13-21-22-23

C.4-9-10-12-13-14-15-20-22-23

D.4-9-11-12-13-14-16-17-19-20-22-23

E.4-9-11-12-13-14-16-18-19-20-22-23

31. 根据下面程序流程图，设计测试用例来完成语句覆盖、判定覆盖、条件覆盖和路径覆盖。

路径：P1:a-c-e-f P2:a-b-e-f P3:a-b-d-f P4:a-c-d-f

判定：M: x>0&&y>0 N: magic<0

条件：M1: x>0 M2: y>0 N: magic<0

语句覆盖：

1.输入：x=0,y=1 输出: magic=0 覆盖路径P4

2.输入：x=1,y=1 输出：magic=12 覆盖路径 P2

判定覆盖：同语句覆盖

条件覆盖：

1.输入：x=0,y=1 输出: magic=0 覆盖路径P4 M1=F,M2=T N=T

2.输入：x=1,y=1 输出：magic=12 覆盖路径 P2 M1=T,M2=T,N=F

3.输入：x=1,y=0 输出：magic=0 覆盖路径 P4 M1=T,M2=F,N=T

路径覆盖：

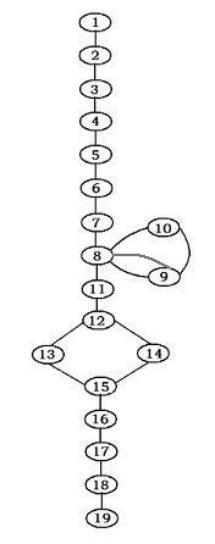
1.输入：x=0,y=1 输出: magic=0 覆盖路径P4

2.输入：x=1,y=1 输出：magic=12 覆盖路径 P2

3.输入：x=12,y=0 输出：magic=2 覆盖路径 P1

4.路径P3不可能实现。

32. 根据给出的某程序流程图：

(1)计算它的环路复杂性；（4）

(2)为了完成基本路径测试，求它的一组独立的路径。

基本路径为：

A：1.2.3.4.5.6.7.8.11.12.14.15.16.17.18.19

B：1.2.3.4.5.6.7.8.9.11.12.14.15.16.17.18.19

C：1.2.3.4.5.6.7.8.910.11.12.13.15.16.17.18.19

D：1.2.3.4.5.6.7.8.9.10.11.12.14.15.16.17.18.19

33. 为以下程序段设计一组测试用例，要求分别满足语句覆盖、判定覆盖、条件覆盖。

int test(int A,int B)

{

if((A>1) AND (B<10)) then

X=A-B;

if((A=2) OR (B>20)) then

X=A+B;

return x;

}

语句覆盖测试用例：A=2，B=0；

判定覆盖测试用例：A=3，B=0；A=2，B=20；

条件覆盖测试用例：A=2，B=0；A=0，B=21；

34. 假设中国某航空公司规定：

中国去欧美的航线所有座位都有食物供应。每个座位都可以播放电影。

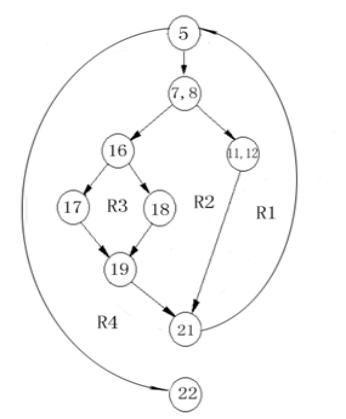
中国去非欧美的国外航线都有食物供应，只有商务仓可以播放电影。

中国国内的航班的商务仓有食物供应，但是不可以播放电影。

中国国内的航班的经济仓仅在飞行时间大于2小时的情况下才有食物供应，但是不可以播放电影。

使用决策表法设计测试用例

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 |
| 航线 | 中国去欧美？ | Y | N | N | N | N | N |
| 中国去非欧美的国外？ | N | Y | Y | N | N | N |
| 中国国内的航班 | N | N | N | Y | Y | Y |
| 商务仓 | -- | Y | N | Y | -- | N |
| 飞行时间>2小时 | -- | -- | -- | -- | Y | -- |
| 服务 | 食物供应 | √ | √ | √ | √ | √ | -- |
| 播放电影 | √ | √ | -- | -- | -- | -- |
| 测试用例 |  | 美国 | 亚洲，商务仓 | 亚洲 | 中国，商务仓 | 中国，飞行间>2 | 中国 |

35. 设一个控制流图如下，请给出环形复杂度和基本测试路径。

(1) 根据程序环形复杂度的计算公式，求出程序路径集合中的独立路径数目。

公式1：V(G)=11-9+2，其中10是控制流图G中边的数量，

8是控制流图中节点的数目。

公式2：V(G)=3+1，其中3是控制流图G中判断节点的数目。

公式3：V(G)=4，其中4是控制流图G中区域的数目。

因此，控制流图G的环形复杂度是4。

（2） 根据上面环形复杂度的计算结果，源程序的基本路径集合中有4条独立路径：

路径1：5->22

路径2：5->7, 8->11, 12->21->5->22

路径3：5->7, 8->16->17->19->21->5->22

路径4：5->7, 8->16->18->19->21->5->22

36. 有一个处理单价为5角钱的饮料的自动售货机软件测试用例的设计。其规格说明如下：若投入5角钱或1元钱的硬币，押下〖橙汁〗或〖啤酒〗的按钮，则相应的饮料就送出来。若售货机没有零钱找，则一个显示〖零钱找完〗的红灯亮，这时在投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则显示〖零钱找完〗的红灯灭，在送出饮料的同时退还5角硬币。

要求：(1) 列出原因和结果，画出因果图;

(2) 根据因果图，建立判定表;

(3) 根据判定表设计测试用例数据。

原因：

1----投5角硬币

2----投1元硬币

3----按【啤酒】按钮

4----按【橙汁】按钮

5----售货机有零钱（灯亮）

6----售货机没有零钱（灯暗）

结果

21----出饮料

22----出饮料，找零钱

23----不出饮料，退钱

因果图如下：

11，12为中间节点

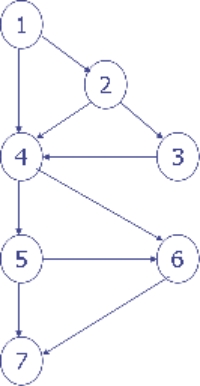
图示

描述已自动生成

（2）（3）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 |
| 条件 | 1 | Y | Y | N | N | N | N |
| 2 | N | N | Y | Y | Y | Y |
| 3 | Y | N | Y | Y | N | N |
| 4 | N | Y | N | N | Y | Y |
| 5 | -- | -- | Y | N | Y | N |
| 6 | -- | -- | N | Y | N | Y |
| 11 | Y | Y | Y | Y | Y | Y |
|  | 12 | Y | Y | Y | Y | Y | Y |
| 动作 | 21 | √ | √ | √ | -- | √ | -- |
| 22 | -- | -- | √ | -- | √ | -- |
| 23 | -- | -- | -- | √ | -- | √ |
| 测试用例 |  | 投5角硬币按啤酒按钮 | 投5角硬币按橙汁按钮 | 投1元硬币选择啤酒按钮售货机有零钱 | 投1元硬币选择啤酒按钮售货机没有零钱 | 投1元硬币选择橙汁按钮售货机有零钱 | 投1元硬币选择橙汁按钮售货机没有零钱 |

37. 使用基本路径测试方法，为以下程序段设计测试用例。

（1）画出程序的控制流图，编号已经给出。

（2）计算程序的循环复杂度，导出程序基本路径集中的独立路径条数。

（3）导出基本路径集，确定程序的独立路径。

（4）根据（3）中的独立路径，设计测试用例

（确保基本路径集中的每一条路径的执行）的输入数据和预期输出。

计算环形复杂度：

10（条边）- 7（个节点）+ 2 = 5

导出独立路径（用语句编号表示）

路径1：1→2→3→4→5→6→7

路径2：1→4→5→6→7

路径3：1→2→4→6→7

路径4：1→2→4→5→7

路径5：1→2→3→4→5→7

测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 用例号 | 路径 | 输入数据  A B X | 预期输出  X |
| TC1 | 1→2→3→4→5→6→7 | 3 0 6 | 3 |
| TC2 | 1→4→5→6→7 | 0 1 3 | 4 |
| TC3 | 1→2→4→6→7 | 2 1 1 | 2 |
| TC4 | 1→2→4→5→7 | 3 1 0 | 0 |
| TC5 | 1→2→3→4→5→7 | 3 0 3 | 1 |

38. 设有一个档案管理系统，要求用户输入以年月表示的日期。假设日期限定在1999年1月~2029年12月，并规定日期由6位数字字符组成，前4位表示年，后2位表示月。现用等价类划分法设计测试用例，来测试程序的"日期检查功能"。

1)划分等价类并编号,下表等价类划分的结果

|  |  |  |
| --- | --- | --- |
| **输入等价类** | **有效等价类** | **无效等价类** |
| 日期的类型及长度 | ①6位数字字符 | ②有非数字字符  ③少于6位数字字符  ④多于6位数字字符 |
| 年份范围 | ⑤在1999~2029之间 | ⑥小于1999  ⑦大于2029 |
| 月份范围 | ⑧在01~12之间 | ⑨等于00  ⑩大于12 |

2)设计测试用例，以便覆盖所有的有效等价类在表中列出了3个有效等价类，编号分别为①、⑤、⑧，设计的测试用例如下：

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖的有效等价类 |
| 200211 | 输入有效 | ①、⑤、⑧ |

为每一个无效等价类设计一个测试用例，设计结果如下：

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖的无效等价类 |
| 99June | 无效输入 | ② |
| 20036 | 无效输入 | ③ |
| 2001006 | 无效输入 | ④ |
| 199712 | 无效输入 | ⑥ |
| 203001 | 无效输入 | ⑦ |
| 200100 | 无效输入 | ⑨ |
| 200113 | 无效输入 | ⑩ |