**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。**

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；

**2. 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。

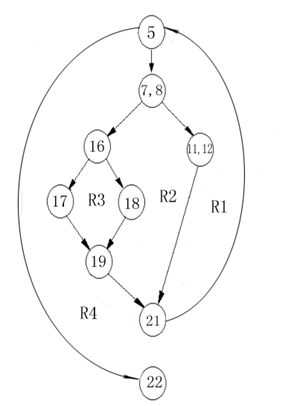
中国国内的航班的商务仓有食物供应，但是不可以播放电影。

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  时  间  >2 | 中国 |

**3. 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

**4. 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

E V

V

∧

∧

∧

（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 |

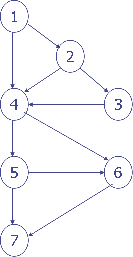
**5、使用基本路径测试方法，为以下程序段设计测试用例。**

**（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 |

**6.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 |