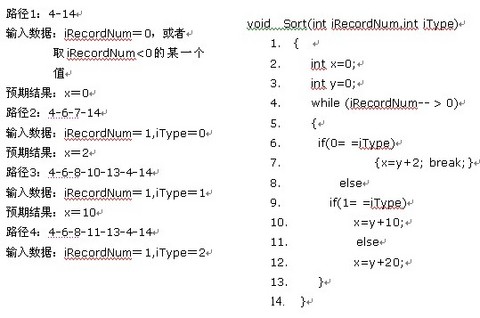
[**白盒测试之基本路径测试法**](http://blog.csdn.net/ccjjyy/article/details/5956769)**实训**

1. **程序代码**

**请按照基本路径测试法的步骤完成如下程序之基本路径测试法实训：**

****

**13**

**11**

**10**

**7**

**14**

**6**

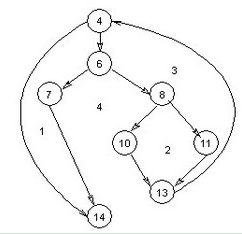
**8**

**4**

**二、画出程序流程图**

流程图用来描述程序控制结构。

**三、画出控制流图**

可将流程图映射到一个相应的流图(假设流程图的菱形决定框中不包含复合条件)。在流图中，每一个圆，称为流图的结点，代表一个或多个语句。一个处理方框序列和一个菱形决测框可被映射为一个结点，流图中的箭头，称为边或连接，代表控制流，类似于流程图中的箭头。一条边必须终止于一个结点，即使该结点并不代表任何语句(例如：if-else-then结构)。由边和结点限定的范围称为区域。计算区域时应包括图外部的范围。

**程序控制流图是：**

**四、计算圈复杂度**

　　圈复杂度是一种为程序逻辑复杂性提供定量测度的软件度量，将该度量用于计算程序的基本的独立路径数目，为确保所有语句至少执行一次的测试数量的上界。独立路径必须包含一条在定义之前不曾用到的边。

　　有以下三种方法计算圈复杂度：

　　流图中区域的数量对应于环型的复杂性;

　　给定流图G的圈复杂度V(G)，定义为V(G)=E-N+2，E是流图中边的数量，N是流图中结点的数量;

　　给定流图G的圈复杂度V(G)，定义为V(G)=P+1，P是流图G中判定结点的数量。

**请计算本程序的圈复杂度：**

1） V(G)=E-N+2=10条边-8个节点+2=4

2）V(G)=P+1=3个判定节点+1=4

**五、分析出独立路径**

没条独立路径是指，和其他的独立路径相比，至少引入一个新处理语句或一个新判断的程序通路。V(G)值正好等于该程序的独立路径的条数。

**程序的独立路径是：**

* 路径1：4-14
* 路径2：4-6-7-14
* 路径3：4-6-8-10-13-4-14
* 路径4：4-6-8-11-13-4-14

**六、准备测试用例**

根据上面的独立路径，去设计输入数据，使程序分别执行到上面四条路径。为了确保基本路径集中的每一条路径的执行，根据判断结点给出的条件，选择适当的数据以保证某一条路径可以被测试到，满足上面例子基本路径集的测试用例是：

**满足上面例子基本路径集的测试用例是：**

路径1：4-14

输入数据：iRecordNum=0,或者取iRecordNum<0的某一个值

预期结果：X=0

路径2：4-6-7-14

输入数据：iRecordNum=1；iType=0

预期结果：x=2

路径3：4-6-8-10-13-4-14

输入数据：iRecordNum=1；iType=1

预期结果：x=10

路径4：4-6-8-11-13-4-14

输入数据：iRecordNum=1；iType=2

预期结果：x=20