实验名称：**结构性测试** 指导教师：

实验日期： 年 月 日 实验地点： 成绩：

实验目的：

1. 掌握结构性测试技术，并能应用结构性测试技术设计测试用例；
2. 对测试用例进行优化设计；

实验内容及基本要求：

实验内容：

**1．题目一：使用逻辑覆盖测试方法测试以下程序段**

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

**{**

1. **int k=0, j=0;**
2. **if ( (x>3)&&(z<10) )**
3. **{**

**4 k=x\*y-1;**

**5       j=sqrt(k);**

**6 }**

**7    if((x==4)||(y>5))**

**8 j=x\*y+10;**

**9    j=j%3;**

**10 }**

说明：程序段中每行开头的数字（1~10）是对每条语句的编号。

（1）画出程序的控制流图（用题中给出的语句编号表示）。

（2）分别以语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、组合覆盖和路径覆盖方法设计测试用例，并写出每个测试用例的执行路径（用题中给出的语句编号表示）。

**2．题目二：选择排序**

下面是选择排序的程序，其中datalist是数据表，它有两个数据成员：一是元素类型为Element的数组V，另一个是数组大小n。算法中用到两个操作，一是取某数组元素V[i]的关键码操作getKey ( )，一是交换两数组元素内容的操作Swap( )：

**void SelectSort ( datalist & list ) {**

/对表list.V[0]到list.V[n-1]进行排序, n是表当前长度。

**for ( int i = 0; i < list.n-1; i++ ) {**

**int k = i;**  //在list.V[i].key到list.V[n-1].key中找具有最小关键码的对象

**for ( int j = i+1; j < list.n; j++)**

**if ( list.V[j].getKey ( ) < list.V[k].getKey ( ) ) k = j;**//当前具最小关键码的对象

**if ( k != i ) Swap ( list.V[i], list.V[k] );**  //交换

**}**

**}**

1. 试计算此程序段的McCabe复杂性；
2. 用基本路径覆盖法给出测试路径；
3. 为各测试路径设计测试用例。

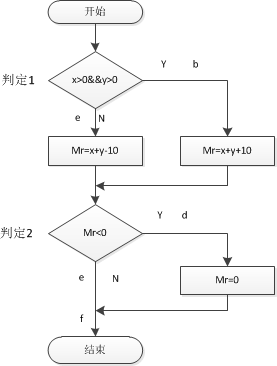
**3．题目三：身份证校验**

用C或java编写一个身份证号码校验程序，并给出相应的白盒测试设计。

4. 题目四：用C或java编写哥德巴赫猜想子程序，任一大于2的偶数，都可表示成两个素数之和， 编写完成该子程序，并给出相应的白盒测试设计。

（或编写 任于大于2的非素数，都可以表示成两个素数之和）

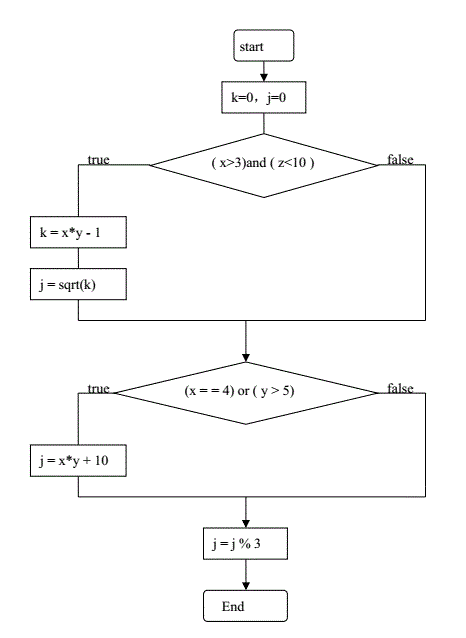
**5. 根据下面的流程图，编写程序，设计用例，使用逻辑覆盖测试该程序。**



基本要求：

1. 根据结构性测试技术设计测试用例，主要考虑逻辑覆盖测试（语句覆盖、判断覆盖、条件覆盖、判断/条件覆盖、条件组合覆盖、路径覆盖）和基路径测试技术；
2. 根据所学知识确定优化策略（原则：用最少的用例检测出更多的缺陷、软件测试的充分性与冗余性考虑），设计两套测试用例集；
3. 根据设计的两套测试用例集进行测试、参照表2所示的缺陷等级给出缺陷列表；
4. 计算测试用例的分支覆盖率、条件覆盖率和语句覆盖率等测试管理指标；
5. 写出实验中遇到的问题及解决的方法

题目一

(1)

（2）语句覆盖

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | X | Y | Z | 条件取值 | 执行路径 |
| 1 | 4 | 5 | 6 | (x>3)&&(z<10)(x==4)||(y>5) | 1 2 3 4 5 6 7 8 9 10 |
| 2 | 2 | 6 | 5 | Y>5 | 1 2 7 9 10 |
| 3 | 5 | 3 | 8 | (x>3)&&(z<10) | 1 2 3 4 5 6 7 9 10 |
| 4 | 4 | 6 | 10 | (x==4)||(y>5) | 1 2 7 8 9 10 |

判定覆盖

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | X | Y | Z | 条件取值 | 执行路径 |
| 1 | 4 | 5 | 6 | （x>3)&&(z<19)(x==4)||(y>5) | 1 2 3 4 5 6 7 8 9 10 |
| 2 | 5 | 3 | 8 | (x>3)&&(z<10) | 1 2 3 4 5 6 7 9 10 |
| 3 | 4 | 6 | 10 | (x==4)||(y>5) | 1 2 7 8 9 10 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | X | Y | Z | 条件取值 | 执行路径 |
| 1 | 5 | 3 | 8 | (x>3)&&(z<10) | 1 2 3 4 5 6 7 9 10 |
| 2 | 4 | 6 | 10 | (x==4)||(y>5) | 1 2 7 8 9 10 |

判定/条件覆盖

①x>3;②z<10;③x==4;④y>5;

①②③④

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | X | Y | Z | 条件取值 | 执行路径 |
| 1 | 4 | 6 | 5 | ①真②真③真④真 | 1 2 3 4 5 6 7 9 10 |
| 2 | 2 | 4 | 13 | ①假②假③假④假 | 1 2 7 8 9 10 |

条件组合覆盖

①x>3;②z<10;③x==4;④y>5;

令x>3为真时，记作T1，即x<=3,记作F1;

令Z<10为真时，记作T2，即z>=10,记作F2;

令x==4为真时，记作T3，即x<>4,记作F3;

令y>5为真时，记作T4，即y<=5,记作F4;

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 具体条件取值 | 覆盖条件 | 判断取值 |
| 1 | X>3,z<10 | T1,T2 | 第一个判定：取真分支 |
| 2 | X>3,z>=10 | T1,F2 | 第一个判定：取假分支 |
| 3 | X<=3,z<=10 | F1,T2 | 第一个判定：取假分支 |
| 4 | X<=3,z>=10 | F1,F2 | 第一个判定：取假分支 |
| 5 | X==4,y>5 | T3,T4 | 第二个判定：取真分支 |
| 6 | X==4,y<=5 | T3,F4 | 第二个判定：取真分支 |
| 7 | X<>4,y>5 | F3,T4 | 第二个判定：取真分支 |
| 8 | X<>4,y<=5 | F3,F4 | 第二个判定：取假分支 |

题目三：代码见打包文件

题目4：

#include <stdio.h>

// 判断是否为素数

int isSuShu(int k){

for (int i =3;i<k;i++){

if(k%i==0)

return -1;

}

// 表明他为素数

return 1;

}

void findTowAdd(int num){

int i=3;

int j=3;

int temp=0;

int isSuShuResult=0;

for(i=3;i<num;i++){

for(j=3;j<=i;j++){

if (i%j==0){

temp=-1;

break;

}

}

if(temp!=-1){

break;

}

isSuShuResult=isSuShu(num-j);

if(isSuShuResult==1){

printf("%d可以分解为两个素数表示,他们是：%d,%d\n",num,j,num-j);

// return ;

}

}

}

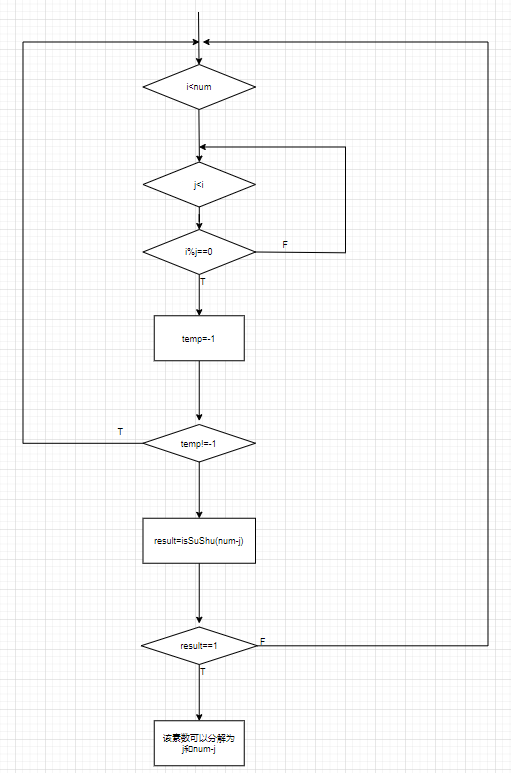
int main(){

findTowAdd(1222234532);

system("pause");

return 0;

}



|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | Num | 条件取值 | 执行路径 |
| 1 | 43532 | I<num&&j<i | 1 2 3 4 5 6 7 8 9 10 |
| 2 | 132 | I<num&&j<i | 1 2 7 9 10 |
| 3 | 935235352 | I<num&&j<i | 1 2 3 4 5 6 7 9 10 |
| 4 | 16736464 | I<num&&j<i | 1 2 7 8 9 10 |