软件测试练习题

刘畅 1753069 谢云鹏 1751920

缪禕天 1753783 杜宝翔 1650273

目录

[1、 三角形问题和万年历问题 2](#_Toc44427751)

[三角形问题： 2](#_Toc44427752)

[万年历问题 4](#_Toc44427753)

[2、佣金问题： 6](#_Toc44427754)

[6、打印机系统 8](#_Toc44427755)

[覆盖策略1（节点覆盖）： 8](#_Toc44427756)

[覆盖策略2（方法覆盖）： 9](#_Toc44427757)

[覆盖策略3（边覆盖） 9](#_Toc44427758)

[7、电信收费问题： 13](#_Toc44427759)

[边界值法： 14](#_Toc44427760)

[等价类法： 14](#_Toc44427761)

[决策表： 16](#_Toc44427762)

[8、销售系统问题： 17](#_Toc44427763)

[语句覆盖 18](#_Toc44427764)

[判断覆盖 19](#_Toc44427765)

[条件覆盖 20](#_Toc44427766)

[判断—条件覆盖 20](#_Toc44427767)

[条件组合覆盖 21](#_Toc44427768)

[9、根据C语言画流程图 21](#_Toc44427769)

# 三角形问题和万年历问题

## 三角形问题：

问题描述：输入三个值，判断能否构成三角形以及三角形的类型

1.1、边界值法：

分析可得问题具有3个输入，分别是三角形的三条边，由于这三条边是互相等价的，我们选择边界值的方法时，可以选择基于单缺陷假设的方法，由于三条边长度可以去无穷大，为了编写程序，我们将三角形的边取值为[1 , 200]，于是我们需要测试超过该范围的测试用例，最终我们选择使用健壮性边界值分析方法：

问题具有三个变量，分别记为a，b，c，于是测试用例有6\*3+1=19种，对于每个变量，我们分别对一个其中一个变量取略小于最小值、最小值、略大于最小值、正常值、略小于最大值、最大值、略大于最大值的7种取值，选取的测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例编号 | a | b | c | 预期输出 |
| 1 | 0 | 100 | 100 | 异常 |
| 2 | 1 | 100 | 100 | 等腰三角形 |
| 3 | 10 | 100 | 95 | 普通三角形 |
| 4 | 100 | 100 | 100 | 等边三角形 |
| 5 | 190 | 100 | 90 | 非三角形 |
| 6 | 200 | 100 | 110 | 普通三角形 |
| 7 | 201 | 100 | 100 | 异常 |
| 8 | 100 | 0 | 100 | 异常 |
| 9 | 100 | 1 | 90 | 非三角形 |
| 10 | 100 | 10 | 105 | 普通三角形 |
| 11 | 100 | 190 | 110 | 普通三角形 |
| 12 | 100 | 200 | 100 | 非三角形 |
| 13 | 100 | 201 | 100 | 异常 |
| 14 | 100 | 100 | 0 | 异常 |
| 15 | 100 | 110 | 1 | 非三角形 |
| 16 | 100 | 100 | 10 | 等腰三角形 |
| 17 | 100 | 95 | 190 | 普通三角形 |
| 18 | 100 | 90 | 200 | 非三角形 |
| 19 | 100 | 110 | 201 | 异常 |

1.2、等价类法：

视三角形的三条边输入同上为a，b，c，取值范围为[0 , 200]

分许输入我们可得，有效等价类有[0 , 200]；无效等价类有小于等于0、大于200

分析输出我们可得出，该程序的输出有：普通三角形、等腰三角形、等边三角形、非三角形、异常。其中前四种为有效的等价类，后一种为无效等价类

接着我们根据输出来分析输入的等价类，我们得到的等价类如下：

1. a+b>c
2. a+c>b
3. b+c>a
4. a=b
5. b=c
6. a=c
7. a=b=c
8. a+b<=c || a+c<=b || b+c<=a

其中，普通三角形对应（1，2，3），等腰三角形对应（1，2，3，4）或（1，2，3，5）或（1，2，3，6），等边三角形对应（1，2，3，4，5，6，7），非三角形对应8

由此我们设计测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例编号 | a | b | c | 预期输出 |
| 1 | 3 | 4 | 5 | 普通三角形 |
| 2 | 3 | 3 | 5 | 等腰三角形 |
| 3 | 3 | 4 | 3 | 等腰三角形 |
| 4 | 3 | 4 | 4 | 等腰三角形 |
| 5 | 3 | 3 | 3 | 等边三角形 |
| 6 | 3 | 4 | 100 | 非三角形 |
| 7 | 3 | 100 | 5 | 非三角形 |
| 8 | 100 | 4 | 5 | 非三角形 |
| 9 | 0 | 4 | 5 | 异常 |
| 10 | 201 | 4 | 5 | 异常 |
| 11 | 3 | 0 | 5 | 异常 |
| 12 | 3 | 201 | 5 | 异常 |
| 13 | 3 | 4 | 0 | 异常 |
| 14 | 3 | 4 | 201 | 异常 |

## 万年历问题

问题描述：输入年月日三个值，输出下一日的日期

2.1、边界值法

分析可得问题具有三个输入，分别是年、月、日

年year的输入范围为[1900 , 2100]、月month的输入范围为[1 , 12]、日day的输入范围为[1 , 31]  
 决定采用普通边界值方法进行测试用例的设计，共计有3\*4+1=13个测试用例，设计的测试用例如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例编号 | year | month | day | 预期输出 |
| 1 | 1900 | 6 | 15 | 1900年6月16日 |
| 2 | 1901 | 5 | 14 | 1901年5月15日 |
| 3 | 2001 | 7 | 18 | 2001年7月19日 |
| 4 | 2099 | 6 | 12 | 2099年6月13日 |
| 5 | 2100 | 5 | 17 | 2100年5月18日 |
| 6 | 2000 | 1 | 15 | 2000年1月16日 |
| 7 | 2002 | 2 | 17 | 2002年2月18日 |
| 8 | 1999 | 11 | 13 | 1999年11月14日 |
| 9 | 2000 | 12 | 15 | 2000年12月16日 |
| 10 | 1998 | 4 | 1 | 1998年4月2日 |
| 11 | 2002 | 5 | 2 | 2002年5月3日 |
| 12 | 2000 | 6 | 30 | 2000年7月1日 |
| 13 | 2003 | 7 | 31 | 2003年8月1日 |

分析可得，边界值方法无法测试2月份随着闰年而变化的天数

2.2、等价类法

将年year划分以下三个有效等价类：

1. 平年
2. 不包括2000年的闰年
3. 2000年

将月month划分为以下4个有效等价类

1. 除12月外有31天的月份（1，3，5，7，8，10月）
2. 有30天的月份（4，6，9，11月）
3. 2月
4. 12月

将天day划分为以下5个等价类：

1、1到27号

2、28号

3、29号

4、30号

5、31号

使用弱一般等价类设计测试用例，用例的个数即为最大等价类数目=5个

测试用例如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | year | month | day | 预期值 |
| 1 | 1993 | 3 | 20 | 1993年3月21日 |
| 2 | 1919 | 9 | 29 | 1919年9月30日 |
| 3 | 2004 | 2 | 28 | 2004年2月29日 |
| 4 | 2008 | 10 | 30 | 2008年10月31日 |
| 5 | 2000 | 12 | 31 | 2001年1月1日 |

出于对平年2月份的考虑，我们需添加额外的测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | year | month | day | 预期值 |
| 6 | 2017 | 2 | 28 | 2017年3月1日 |

# 2、佣金问题：

问题描述：电脑销售系统，主机（25￥单位价格，每月最多销售的数量为70），显示器（30￥单位价格，每月最多销售数量为80），外设（45￥单位价格，每月最多销售的数量为90）；每个销售员每月至少销售一台完整的机器，当系统的主机这个变量接受到-1值的时候，系统自动统计该销售员本月的销售总额。当销售额小于等于1000（包括1000）按照10%提佣金，当销售额在1000-1800之间（包括1800）的时候按照15%提佣金，当销售额大于1800时按照20%提佣金。用边界值法设计测试用例。

使用边界值法进行测试用例的设计

我们假设主机、显示器、外设的销售额分别记为a，b，c

由问题描述可得知，，，

使用健壮性边界值分析方法可得，一共有6\*3+1=19个测试用例

a取0，1，3，35，68，70，71

b取0，1，5，40，75，80，81

c取0，1，7，45，83，90，91

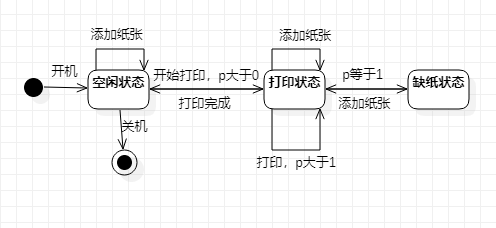
设计的测试用例如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | a | b | c | 预期值 |
| 1 | 0 | 40 | 45 | 异常 |
| 2 | 1 | 40 | 45 | 650 |
| 3 | 3 | 40 | 45 | 660 |
| 4 | 35 | 40 | 45 | 820 |
| 5 | 68 | 40 | 45 | 985 |
| 6 | 70 | 40 | 45 | 995 |
| 7 | 71 | 40 | 45 | 异常 |
| 8 | 35 | 0 | 45 | 异常 |
| 9 | 35 | 1 | 45 | 586 |
| 10 | 35 | 5 | 45 | 610 |
| 11 | 35 | 75 | 45 | 1030 |
| 12 | 35 | 80 | 45 | 1060 |
| 13 | 35 | 81 | 45 | 异常 |
| 14 | 35 | 40 | 0 | 异常 |
| 15 | 35 | 40 | 1 | 424 |
| 16 | 35 | 40 | 7 | 478 |
| 17 | 35 | 40 | 83. | 1162 |
| 18 | 35 | 40 | 90 | 1225 |
| 19 | 35 | 40 | 91 | 异常 |

3，4，5略

# 6、打印机系统

使用transition tree的方法。首先根据上课内容绘制打印系统的状态图：



## 覆盖策略1（节点覆盖）：

1. 开机
2. 开始打印
3. 打印直到缺纸状态
4. 添加纸张
5. 打印
6. 打印完成
7. 关机

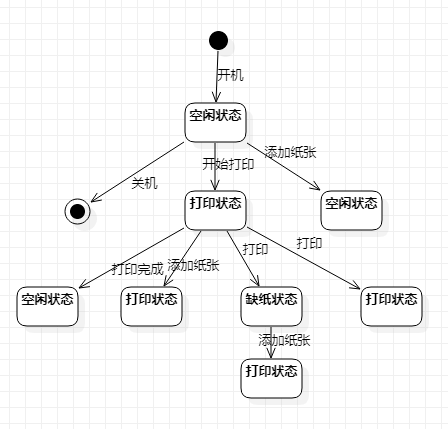
## 覆盖策略2（方法覆盖）：

1. 开机
2. 开始打印
3. 打印到缺纸状态
4. 添加纸张
5. 关机

## 覆盖策略3（边覆盖）

1. 开机
2. 添加纸张
3. 开始打印
4. 添加纸张
5. 打印直到缺纸状态
6. 添加纸张
7. 打印直到完成
8. 关机

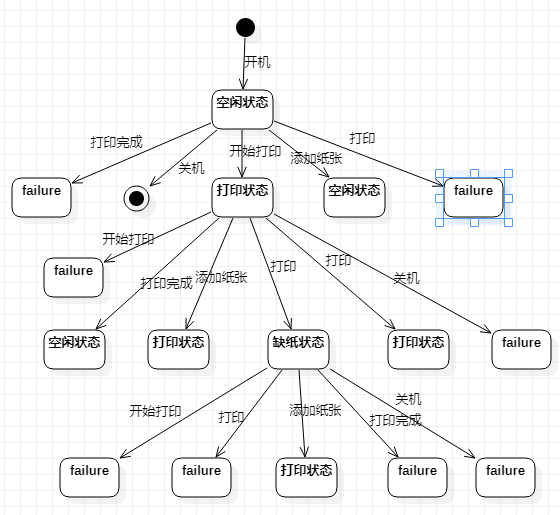
使用转化算法将状态图转化为树状图如下：



可以看出，该数有6个叶子节点，为了覆盖所有的叶子节点我们需要设计6个测试用例，设计的测试用例如下：

|  |  |
| --- | --- |
| 测试用例编号 | 活动 |
| 1 | 1. 开机 2. 关机 |
| 2 | 1. 开机 2. 添加纸张 |
| 3 | 1. 开机 2. 开始打印 3. 打印完成 |
| 4 | 1. 开机 2. 开始打印 3. 添加纸张 |
| 5 | 1. 开机 2. 开始打印 3. 打印 4. 添加纸张 |
| 6 | 1. 开机 2. 开始打印 3. 打印 |

接下来进行健壮性的考虑，对每个节点添加缺少的非法功能来达到failure状态



可以看出，该数有14个叶子节点，为了覆盖所有的叶子节点我们需要设计14个测试用例，设计的额外测试用例如下：

|  |  |
| --- | --- |
| 测试用例编号 | 活动 |
| 7 | 1. 开机 2. 打印完成 |
| 8 | 1. 开机 2. 开始打印 3. 开始打印 |
| 9 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 开始打印 |
| 10 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 打印 |
| 11 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 打印完成 |
| 12 | 1. 开机 2. 开始打印 3. 打印到缺纸状态 4. 关机 |
| 13 | 1. 开机 2. 开始打印 3. 关机 |
| 14 | 1. 开机 2. 打印 |

# 7、电信收费问题：

问题描述：研究一个与我们的生活息息相关的电信收费问题系统，需求描述如下：  
每月的电话总费用=基本月租费+折扣后的实际的通话费，如果没有折扣则按实际通话费计算，基本月租费为25元，每分钟通话费为0.15元。  
实际通话费是否有折扣与当月的通话时间（分钟）和本年度至本月的累计未按时缴费的次数有关。跨年度未交费与折扣无关，但跨年未交部分每月需要交付总额5%的滞纳金。  
当月的通话分钟数和折扣比例及本年度未按时缴费次数之间有直接的对应关系，如果本年度的未按时缴费的次数超过本月通话时间所对应的容许值则免于折扣，并按实际的通话费计算。

电话费的收取，采用在线支付，支付方式为：支付宝或银行卡（开发简单的模拟子系统），支付后打印支付成功或不成功清单。

通话时间和折扣比例及未按时缴费次数的关系为：

本月通话的分钟数 通话时间段的最大容许不按时缴费次数 通话时间段的折扣率  
0＜ 通话时间≤60 1 1.0％  
60＜ 通话时间≤120 2 1.5％  
120＜ 通话时间≤180 3 2.0％  
180＜ 通话时间≤300 3 2.5％  
300＜ 通话时间 6 3.0％

分别用边界值、等价类和决策表设计测试用例，并综合分析得出合理的测试用例集。

## 边界值法：

首先分析可得输入变量有两个：通话时间和不按时缴费次数，分别记为N和M。我们可分析得知，N的有效取值范围为0<N<44640（每月31天计，每月共44640分钟），M的有效取值范围为0到11的整数。

根据边界值区间，使用健壮边界值分析方法设计测试用例：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | N | M | 预期结果 |
| 1 | -10 | 5 | 异常 |
| 2 | 0 | 4 | 25 |
| 3 | 10 | 6 | 26.5 |
| 4 | 200 | 5 | 55 |
| 5 | 44630 | 7 | 6719.5 |
| 6 | 44640 | 3 | 6520.12 |
| 7 | 44655 | 6 | 异常 |
| 8 | 210 | -1 | 异常 |
| 9 | 208 | 0 | 55.42 |
| 10 | 208 | 1 | 55.42 |
| 11 | 200 | 10 | 55 |
| 12 | 211 | 11 | 56.65 |
| 13 | 200 | 12 | 异常 |

## 等价类法：

对于上述输入变量N，M。根据题目分析可划分为如下等价类：

变量N：

1、

2、

3

4、

5、

变量M：

1、{0，1}

2、{2}

3、{3}

4、{4，5，6}

5、{7，8，9，10，11}

使用强一般等价类方法进行测试用例的设计：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | N | M | 预期结果 |
| 1 | 30 | 0 | 29.455 |
| 2 | 30 | 2 | 29.5 |
| 3 | 30 | 3 | 29.5 |
| 4 | 30 | 5 | 29.5 |
| 5 | 30 | 10 | 29.5 |
| 6 | 80 | 0 | 36.4 |
| 7 | 80 | 2 | 36.4 |
| 8 | 80 | 3 | 37 |
| 9 | 80 | 5 | 37 |
| 10 | 80 | 9 | 37 |
| 11 | 160 | 1 | 48.52 |
| 12 | 160 | 2 | 48.52 |
| 13 | 160 | 3 | 48.52 |
| 14 | 160 | 6 | 49 |
| 15 | 160 | 8 | 49 |
| 16 | 200 | 0 | 54.25 |
| 17 | 200 | 2 | 54.25 |
| 18 | 200 | 3 | 54.25 |
| 19 | 200 | 5 | 55 |
| 20 | 200 | 7 | 55 |
| 21 | 20000 | 0 | 2935 |
| 22 | 20000 | 2 | 2935 |
| 23 | 20000 | 3 | 2935 |
| 24 | 20000 | 6 | 2935 |
| 25 | 20000 | 10 | 3025 |

## 决策表：

对上述变量N和M的等价类进行编号：

和

使用该等价类进行决策表的设计：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 标号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  | |  | |  | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |
| 有折扣 | **X** |  | **X** |  | **X** |  | **X** |  | **X** |  |
| 无折扣 |  | **X** |  | **X** |  | **X** |  | **X** |  | **X** |
| 折扣 | 1% |  | 1.5% |  | 2% |  | 2.5% |  | 3% |  |

根据决策表设计的测试用例如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | N | M | 预期输出 |
| 1 | 25 | 0 | 28.7125 |
| 2 | 25 | 5 | 28.75 |
| 3 | 80 | 1 | 36.82 |
| 4 | 80 | 7 | 37 |
| 5 | 150 | 3 | 47.05 |
| 6 | 150 | 9 | 47.5 |
| 7 | 220 | 3 | 57.175 |
| 8 | 220 | 7 | 58 |
| 9 | 500 | 6 | 97.75 |
| 10 | 800 | 11 | 145 |

为了得出最优的测试集，我们分析得使用边界值法得出得测试用例无法覆盖到所有的情况，而决策表法得出的测试用例没有考虑到边界的情况，我们综合两种方法取边界值法中的1，2，6，7，8，12，13号测试用例和决策表中所有的测试用例形成的测试用例集。

# 8、销售系统问题：

问题描述：

如果销售员的年销售额大于200万RMB且请假天数不超过10天的情况下，现金到帐大于等于60%，则佣金（提成）系数为7，即佣金值为销售额除以佣金系数；现金到帐小于60%，佣金不予计算。所有其他情况且现金到帐小于等于85%，则按佣金系数均为6计算佣金，现金到账大于85%，佣金系数按5处理。

根据题意设计流程图并设计测试用例实现白盒测试（White Box Test）的1）语句覆盖，2）判断覆盖，3）条件覆盖，4）判断—条件覆盖，5）条件组合覆盖（测试用例及覆盖表示要清晰）。

问题解决：

依据题意我们编写出如下的代码用于白盒测试：

1 public class Commission {  
2 public double calculate(int sales, int absence, double percent){  
3 if(sales>2000000 && absence<= 10 && percent>=0.6){  
4 return sales/7;  
5 }  
6 else if (percent<0.6){  
7 return 0;  
8 }  
9 else if(percent<=0.85){  
10 return sales/6;  
11 }  
12 else{  
13 return sales/5;  
14 }  
15 }  
16}

## 语句覆盖

基于语句覆盖设计的测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 年销售额sales | 请假天数absence | 现金到账百分比percent | 预期输出 |
| 1 | 2800000 | 5 | 0.8 | 400000.0 |
| 2 | 1000000 | 6 | 0.5 | 0 |
| 3 | 3600000 | 12 | 0.65 | 600000.0 |
| 4 | 5000000 | 15 | 0.9 | 1000000.0 |

测试用例1覆盖到语句3，4

测试用例2覆盖到语句6，7

测试用例3覆盖到语句9，10

测试用例4覆盖到语句12，13

该测试集覆盖到了所有的有效语句

## 判断覆盖

基于判断覆盖设计的测试用例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 年销售额sales | 请假天数absence | 现金到账百分比percent | 预期输出 |
| 1 | 2800000 | 5 | 0.8 | 400000.0 |
| 2 | 1000000 | 6 | 0.5 | 0 |
| 3 | 3600000 | 12 | 0.65 | 600000.0 |
| 4 | 5000000 | 15 | 0.9 | 1000000.0 |

我们发现，基于判断覆盖的测试用例与基于语句覆盖的测试用例相同，因为该测试用例同时能满足基于判断的覆盖

测试用例1覆盖判断语句3的真

测试用例2覆盖判断语句3的假和判断语句6的真

测试用例3覆盖判断语句3的假、判断语句6的假和判断语句9的真

测试用例4覆盖判断语句3的假、判断语句6的假和判断语句9的假

## 条件覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 年销售额sales | 请假天数absence | 现金到账百分比percent | 预期输出 |
| 1 | 2800000 | 5 | 0.8 | 400000.0 |
| 2 | 1000000 | 15 | 0.5 | 0 |
| 3 | 1500000 | 12 | 0.7 | 600000.0 |
| 4 | 1500000 | 12 | 0.9 | 1000000.0 |

测试用例1和测试用例2满足了判断语句3中所有子表达式的真假情况

测试用例2满足了判断语句6中子表达式的真情况

测试用例3满足了判断语句6子表达式的假情况、判断语句9中子表达式的真情况

测试用例4满足了判断语句9中子表达式的假情况

## 判断—条件覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 年销售额sales | 请假天数absence | 现金到账百分比percent | 预期输出 |
| 1 | 2800000 | 5 | 0.8 | 400000.0 |
| 2 | 1000000 | 15 | 0.5 | 0 |
| 3 | 1500000 | 12 | 0.7 | 600000.0 |
| 4 | 1500000 | 12 | 0.9 | 1000000.0 |

该测试用例集与条件覆盖测试用例集相同，即满足了条件覆盖

测试用例1覆盖判断语句3的真

测试用例2覆盖判断语句3的假和判断语句6的真

测试用例3覆盖判断语句3的假、判断语句6的假和判断语句9的真

测试用例4覆盖判断语句3的假、判断语句6的假和判断语句9的假

于是该测试用例同时满足判断、条件覆盖

## 条件组合覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 年销售额sales | 请假天数absence | 现金到账百分比percent | 预期输出 |
| 1 | 2800000 | 8 | 0.7 | 400000.0 |
| 2 | 1000000 | 12 | 0.5 | 0 |
| 3 | 1200000 | 8 | 0.7 | 200000.0 |
| 4 | 1200000 | 12 | 0.8 | 200000.0 |
| 5 | 1500000 | 6 | 0.5 | 0 |
| 6 | 3000000 | 8 | 0.5 | 0 |
| 7 | 3000000 | 12 | 0.9 | 600000.0 |
| 8 | 3000000 | 12 | 0.5 | 0 |

首先8个测试用例覆盖了判断1(1个子表达式)的真假，然后测试用例2覆盖了判断2(1个子表达式)的真，测试用例3覆盖了判断2的假与判断3(1个子表达式)的真，测试用例7覆盖了判断3(1个子表示)的假。

# 9、根据C语言画流程图

将所有的for语句看成while语句

条件如下：

a1: \*Str & 0x80

b1: \*(Str + 1) & 0x80

c1: i >= 24

d1: j >= 3

e1: k >= 8

f1: Buf [i \* 3 + j]>>(7-k) & 1

