第一章

一、选择题

1. 缺陷产生的原因包括（）
   A. 交流不充分及沟通不畅；软件需求的变更；软件开发工具的缺陷
   B. 软件的复杂性；软件项目的时间压力
   C. 程序开发人员的错误；软件项目文档的缺乏
   D. 以上都是
2. 下面有关软件缺陷的说法中错误的是（）
   A. 缺陷就是软件产品在开发中存在的错误
   B. 缺陷就是软件维护过程中存在的错误、毛病等各种问题
   C. 缺陷就是导致系统程序崩溃的错误
   D. 缺陷就是系统所需要实现某种功能的失效和违背
3. 以下选项不属于软件缺陷的是（）
   A. 软件没有实现产品规格说明所要求的功能
   B. 软件中出现了产品规格说明不应该出现的功能
   C. 软件实现了产品规格没有提到的功能
   D. 软件满足用户需求，但测试人员认为用户需求不合常理
4. 下面有关测试原则的说法正确的是（）
   A. 测试用例应由测试的输入数据和预期的输出结果组成
   B. 测试用例只需选取合理的输入数据
   C. 软件最好由开发该软件的程序员自己来做测试
   D. 使用测试用例进行测试是为了检查程序是否做了它该做的事
5. 在软件生命周期的哪一个阶段，软件缺陷修复费用最低（）
   A. 需求分析(编制产品说明书)　　B. 设计
   C. 编码　　D. 产品发布
6. 为了提高测试的效率，应该（）
   A. 随机地选取测试数据
   B. 取一切可能的输入数据作为测试数据
   C. 在完成编码以后制定软件的测试计划
   D. 选择发现错误可能性大的数据作为测试数据
7. 下列说法不正确的是（）
   A. 测试不能证明软件的正确性
   B. 测试员需要良好的沟通技巧
   C. QA 与 testing 属于一个层次的概念
   D. 成功的测试是发现了错误的测试
8. 下列（）不属于软件缺陷。
   A. 测试人员主观认为不合理的地方
   B. 软件未达到产品说明书标明的功能
   C. 软件出现了产品说明书指明不会出现的错误
   D. 软件功能超出产品说明书指明范围
9. 产品发布后修复软件缺陷比项目开发早期这样做的费用要高（）
   A. 1~2 倍　　B.10-20 倍　　C. 50 倍　　D. 100 倍或更高

10. 软件测试的目的是（）

A. 发现程序中的所有错误

B. 尽可能多地发现程序中的错误

C. 证明程序是正确的

D. 调试程序

11.经验表明，在程序测试中，某模块与其他模块相比，若该模块已发现并改正的错误较多，则该模块中残存的错误数目与其他模块相比，通常应该（）

A.较少　　B. 较多　　C. 相似　　D. 不确定

12.导致软件缺陷的最大原因是（）

A. 需求分析　　B. 设计　　C. 编码　　D. 测试

13. 下列中不属于测试原则的是（）

A. 软件测试是有风险的行为

B. 完全测试程序是不可能的

8. 测试无法显示潜伏的软件缺陷

D.找到的缺陷越多软件的缺陷就越少

14. 一个成功的测试是（）

A. 发现错误码　　B. 发现了至今尚未发现的错误

C. 没有发现错误码　　D.证明发现不了错误

15. 权衡多个因素，较实用的软件测试停止标准是（）

A. 测试超过了预定时间，则停止测试。

B. 根据查出的缺陷总数量决定是否停止测试。

C. 测试成本超过了预期计划，则停止测试。

D. 分析发现的缺陷数量和测试投入成本曲线图，确定应继续测试还是停止测试。

16.第一类测试方法与第二类测试方法的本质区别体现在（）

A. 执行测试的人员不同　　B. 执行测试的时间不同

C. 执行测试的目的不同　　D. 执行测试的效果不同

17.下列不属于软件缺陷的是（）

A. 银行 POS 机在用户取款时翻倍吐钱，取 100 吐 200

B. 计算机病毒发作，屏幕出现熊猫烧香画面

C. 网上售票软件反应迟钝，用户难以正常买票

D. 某软件在进行修改升级之后， 原来正常的功能现在出错了

二、填空题

1. 软件测试是使用人工或自动的手段来_运行_或_测定_某个软件系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。

2. 软件质量成本包括所有由质量工作或者进行与质量有关的活动所导致的成本，包括_预防成本_、评价成本、失效成本。

3. 软件缺陷产生的原因包括_软件自身的特点、_团队合作、_技术和实现问题_以及_管理问题_等

4. 软件缺陷就是存在于软件(文档、数据、程序)之中的那些不希望或不可接受的偏差。它的存在会导致软件产品在某种程度上不能满足用户的需要。

三、判断题

1、没有可运行的程序，就无法进行任何测试工作。 ✘

2、软件测试针对的是初级程序员编写的程序，资深程序员编写的程序无需测试。 ✘

3、测试是为了验证软件已正确地实现了用户的要求。 ✘
4、测试一个程序，只需按程序的预期工作方式运行它就行了。 ✘
5、好的测试员坚持不懈追求完美。 ✘
6、软件测试工具可以代替软件测试员。 ✘
7、在软件开发过程中，若能推迟暴露其中的错误，则为修复和改进错误所花费的代价就会降低。 ✘
8、程序员与测试工作无关。 ✘
9、我是个很棒的程序员，我无需进行单元测试。✘
10、软件缺陷是导致软件失效的必要，而非充分要求。✔
11、在软件产品计划阶段，不必进行 SQA 活动。✘

四、解答题

1、什么是软件测试、软件质量保证?分析它们之间的关系如何。

软件测试：使用人工或自动手段来运行或测定某个软件系统过程，检验它是否满足规定的需求或弄清预期结果与实际结果之间差别。

软件质量保证：SQA 是为保证软件产品和服务充分满足用户要求的质量而进行的有计划、有组织的活动。

关系：

①软件测试是事后检查，SQA 是贯穿于整个过程。
②SQA 侧重于过程的管理和控制，是一项管理工作。
③软件测试是过程管理和控制策略的具体执行。是一项技术型工作。
④有了 SQA，测试工作就可以被客观的检查评价。
⑤软件测试为 SQA 提供数据和依据。
⑥测试通常被认为是质量控制的主要手段。

2.试分析应如何降低软件质量成本。

软件质量成本包括：预防成本，评估成本，失败成本。

预防成本、评价成本的合理变化区间范围较小，而失败成本的变化范围非常大，小到可以忽略不计，大到无法承受。

如果不投入必要的预防成本、评价成本，那么软件之力将没有保障，可能问题很多，投入实际使用后可能产生软件失败，导致严重后果和重大损失。

所以软件生产应当投入合理的预防成本和评价成本，提高软件质量，防止软件失败，降低失败成本，从而降低总的软件质量成本。

3、什么是 PIE 模型?试分析 PIE 模型对软件测试设计有何指导意义。

PIE 是 Propagation、Infection、Execution 三个英文字母首字母的缩写。程序中的缺陷，如果要通过动态测试来观察到，需要三个必要的条件：

（1）程序执行路径必须通过错误的代码（Execution-执行）；
（2）在执行错误代码的时候必须符合某个或者某些特定条件，从而触发出错误的中间状态（Infection-感染）；
（3）错误的中间状态必须传播到最后输出，使得观测到输出结果与预期结果不一致（Propagation-传播）

这就是 PIE 模型。

对某个软件进行软件测试时，如果包含缺陷 Fault 的代码可能没有被执行到；或者测试执行到了包含缺陷 Fault 的代码，但由于不满足待定的输入条件，没有产生错误的中间状态 error；或者产生了错误的中间状态，但没有传播到最后输出，从外部没有发现问题，以上情况都会导致测试工作不充分，发现不了软件中存在的缺陷。

PIE 模型对软件测试设计的指导意义在于：通过执行软件，能够发现的问题只有 PIE 模型中外部层面的软件失败 Failure，也就是表现出来的问题。程序中处于内部静态层次的缺陷 Fault，和内部中间状态层次的错误 Error，是难以通过执行软件来直接检测出来的。测试设计要做的重要工作之一，就是如何恰当的设计测试数据，使得可能存在的软件缺陷 Fault，通过程序执行都尽可能的产生失败 Failure 并被外部观察到。

4.试分析软件缺陷产生的原因。

①软件自身特点，软件需求不明确，软件结构复杂，精确时间同步不准确，运行环境复杂。

②项目管理，开发人员与用户，或开发人员间沟通不够，开发人员理解不同，技术人员水平参差不齐。

③团队合作，系统结构设计不合理，没有备份，程序逻辑路径或数据范围不够，算法，语法，计算错误。

④技术问题，缺乏质量意识，流程不够完善，软件文档不完善，开发过程不按照规定。

5.试分析为什么要对软件进行质量保证与测试。

①及早发现问题，解决问题，降低返工和修复缺陷的版本。

②防止事故发生，降低失败成本。

③保证软件产品达到一定的质量标准

④对软件质量进行客观的评价。

⑤提高软件产品质量，满足用户需求。

6.计算机病毒是否是软件缺陷?为什么?

不属于，软件缺陷是指软件中存在的偏差，且病毒则来自外部。

7.第一类测试方法与第二类测试方法各自的优缺点是什么?

以正向思维方式，针对软件系统的所有功能点，逐个验证其正确性，被称为第一类软件测试方法。

以逆向思维方式，去发现软件中可能存在的各种问题，验证软件是“不工作的”，被称为第二类软件测试方法。

第一类测试方法测试要求就是软件的规格说明，简单明确，易于实施，但不利于发现软件中的问题。

第二类测试方法需要测试各种可能的情况，包括特殊情况、异常状况等，测试要求更多，但有利于发现软件中可能存在的问题。

8.针对以下代码，分析代码中存在的问题和缺陷。

```
public class getScoreAverage
{    public float getAverage( int [] scores )
     {    if (scores==null || scores.length==0)
          {        throw new NullPointerException();
          }
          float sum = 0.0F;
          int j=scores.length;
          for (int i=l; i<j; 1++)
          {      sum += scores[i];
          }
          return sum/j;
     }
}
```

①如果成绩数组为空或者长度为 0，应给出具体的提示信息。

②循环控制变量 i 的初值应为 0。

③这段代码没有注释，规范的代码应当有良好的注释。

9.有程序段如下:

```
public int get_ max(int x,int y,int z){
    int max;
    if(x>=y)
    {    max  =  x;   }
    else
    {    max  =  y;   }
    if(  z>=x   )
    {    max  =  z;   }
         return max;          }
```

(1)试分析该程序段有何逻辑错误。

程序的逻辑错误在于，z 只与 x 进行了比较，而没有与 y 进行比较。当 z 与 x 进行比较，且 z>x 时，程序就会让 max = z，但此时如果 z<y，就出错了。

(2) 设计 1 个测试数据，使执行该测试时会执行到缺陷代码但不会触发错误。

x=9,y=8,z=7(z<x 即可)

(3) 设计 1 个测试数据，使执行该测试时会执行到缺陷代码并触发错误，但不会引起失败。

x=9,y=8,z=10(z 最大即可)

(4) 设计 1 个测试数据，使执行该测试时会执行到缺陷代码，触发错误，并引起失败。

x=9,y=11,z=10(x<y,且 z>x 即可)

第二章

一、选择题

1 软件测试技术可以分为静态测试和动态测试，下列说法中错误的是（ ）。

A、静态测试是指不运行程序，通过检查和阅读等手段来发现程序中的错误。

B、动态测试是指实际运行程序，通过运行的结果来发现程序中的错误。

C、动态测试包括黑盒测试和白盒测试。

D、白盒测试是静态测试，黑盒测试是动态测试。

2 划分软件测试属于白盒测试还是黑盒测试的依据是 （ ）

A、是否执行程序代码

B、是否能看到软件设计文档

C、是否能看到被测源程序

D、运行结果是否确定

3（ ）把黑盒测试和白盒测试的界限打乱了。

A、灰盒测试 B、动态测试 C、静态测试 D、失败测试

4 在软件测试用例设计的方法中，最常用的方法是黑盒测试和白盒测试，其中不属于白盒测试所关注的是（ ）

A、程序结构 B、软件外部功能 C、程序正确性 D、程序内部逻辑

5 下列哪项不属于黑盒测试的优点（ ）。

A. 不需要源代码

B. 测试简单易行

C. 可以对代码进行有针对性的测试

D. 可以发现软件功能上的问题

二、填空题

1、动态测试的两个基本要素是_被测试程序_、__测试用例。
2、软件测试的 W 模型由两个 V 字组成，分别代表_开发_与_测试__过程。
3、按照是否需要知道被测试程序的内部结构，测试方法可以分为：黑盒测试和白盒测试。

三、判断题

1、黑盒测试的测试用例是根据程序内部逻辑设计的。（× ）
2、软件测试是有效的发现软件缺陷的手段。（ √ ）
3、集成测试计划在需求分析阶段末提交。（× ）

四、解答题

1、试对比分析软件测试的 V 模型和 W 模型。
2、试分析黑盒测试、白盒测试、静态测试、动态测试之间的关系。
　黑盒测试都是动态测试。白盒测试有动态测试也有静态测试。动态测试既可能是黑盒测试，也可能是白盒测试。静态测试只能是白盒测试。
3、试对比分析黑盒测试、白盒测试各自的优缺点。

	优点	缺点
黑盒测试	不需要源代码 较为简单易行	没有代码覆盖的针对性 发现缺陷迟
白盒测试	发现缺陷早 降低返工成本 覆盖关键代码 发现缺陷概率高	非常耗费时间 需要知识和经验积累 技术能力要求高 准备工作多

4、 你认为应如何对一个软件实施测试，试结合你所参与过的软件项目，阐述软件测试工作的一般过程。
（1）分析明确测试需求
（2）制定测试计划
（3）进行测试设计
（4）测试开发
（5）测试执行和记录
（6）测试总结和评价
5、试分析动态白盒测试与黑盒测试的区别。
两者测试用例设计的依据是不同的。
动态白盒测试的测试用例设计依据是程序的内部逻辑结构。
而黑盒测试的测试用例设计依据是程序的规格说明书。

第三章

一、选择题

1 凭经验或直觉推测可能的错误，列出程序中可能有的错误和容易发生错误的特殊情况，选择测试用例的测试方法叫（)
A 等价类划分 B 边界值分析 C 错误推测法 D 逻辑覆盖测试
2 黑盒测试技术中不包括（ ）。

A 等价类划分 B 边界值分析 C 错误推测法 D 逻辑覆盖

3 黑盒测试技术，使用最广的用例设计技术是（ ）

A 等价类划分 B 边界值分析 C 错误推测法 D 逻辑覆盖

4 在某大学学籍管理信息系统中，假设学生年龄的输入范围为 16-40，则根据黑盒测试中的等价类划分技术，下面划分正确的是（ ）。

A 可划分为 2 个有效等价类，2 个无效等价类

B 可划分为 1 个有效等价类，2 个无效等价类

C 可划分为 2 个有效等价类，1 个无效等价类

D 可划分为 1 个有效等价类，1 个无效等价类

5 有一组测试用例使得被测程序的每一个分支至少被执行一次，它满足的覆盖标准是（ ）。

A 语句覆盖 B 判定覆盖 C 条件覆盖 D 路径覆盖

6 在确定黑盒测试策略时，优先选用的方法是 （）

A 边界值分析法 B 等价类划分 C 错误推断法 D 决策表方法

7（ ）方法根据输出对输入的依赖关系设计测试用例。

A 路径测试 B 等价类 C 因果图 D 归纳测试

8 对于参数配置类的软件，要用（ ）选择较少的组合方式达到最佳效果。

A 等价类划分 B 因果图法 C 正交试验法 D 场景法

9 对于业务流清晰的系统可以利用（ ）贯穿整个测试用例设计过程并在用例中综合使用各种测试方法。

A 等价类划分 B 因果图法 C 正交试验法 D 场景法

10 下列不属于黑盒测试方法的是( )。

A 等价类划分 B 因果图 C 边界值分析 D 变异测试

11 用边界值分析法，假定 1<X<100，那么整数 X 在测试中应取的边界值不包括（ ）。

A、X=1，X=100；

B、X=0，X=101；

C、X=2，X=99；

D、X=3，X=98；

二、填空题

1、等价类划分有两种不同的情况：有效等价类和无效等价类。

2、如果有多个输入条件，并且各个条件之间存在关联，那么仅仅只是覆盖所有的等价类还不够，还需要考虑等价类之间的_组合__。

3、各个被测变量的等价类总数等于其__有效等价类总数_ 加上 __无效等价类总数__。

三、判断题

1、一个测试用例可覆盖多个有效等价类和无效等价类。（×）

2、不同的等价类划分得到的测试用例的质量不同。（ √ ）

3、强健壮等价类测试中测试用例个数为各个被测变量的等价类总数的和。（×）

注:还有各个被测变量的无效等价类数之和

四、解答题

1、某种信息加密代码由三部分组成，这三部分的名称和内容如下。

（1）加密类型码：空白或三位数字；

（2）前缀码 ：非’0’或’1’开头的三位数；
（3）后缀码 ：四位数字。
假定被测试的程序能接受一切符合上述规定的信息加密代码，拒绝所有不符合规定的信息加密代码，试用等价类划分法，分析它所有的等价类，并设计测试用例。

等价类划分表		
输入类型	有效等价类	
加密类型码	1	空白
	2	三位数字
前缀码	3	非0开头的三位数
		非1开头的三位数
后缀码	4	四位数字
	无效等价类	
加密类型码	5	非空白且有非数字字符
	6	少于三位数字
	7	多于三位数字
前缀码	8	少于三位数字
	9	多于三位数字
	10	0开头的三位数
	11	1开头的三位数
	12	有非数字字符
后缀码	13	有非数字字符
	14	少于四位数字
	15	多于四位数字

测试用例表					
用例编号	输入类型			预期输出	覆盖等价类
	加密类型码	前缀码	后缀码		
1		123	4567	接受(有效)	134
2	123	805	9876	接受(有效)	234
3	20A	123	4567	拒绝(无效)	5
4	33	234	5678	拒绝(无效)	6
5	1234	234	4567	拒绝(无效)	7
6	123	2B3	1234	拒绝(无效)	8
7	123	0'13	1234	拒绝(无效)	9
8	123	123	1234	拒绝(无效)	10
9	123	23	1234	拒绝(无效)	11
10	123	2345	1234	拒绝(无效)	12
11	123	234	1B34	拒绝(无效)	13
12	123	234	34	拒绝(无效)	14
13	123	234	23345	拒绝(无效)	15

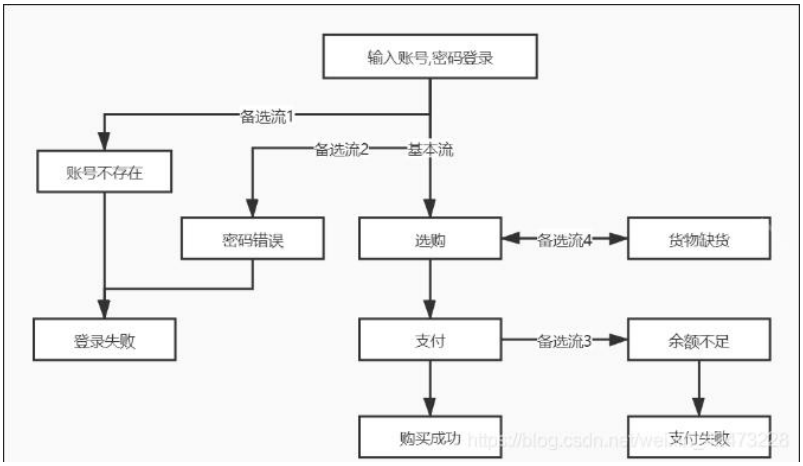
2、某“银行网站系统”登录界面如下图所示，试采用错误推测法，举出 10 种常见问题或错误，并设计 10 个测试用例。

常见问题或错误表		
输入类型	序号	问题
账号	1	未填写账号
	2	账号位数过长
	3	账号位数过短
	4	账号出现非法字符
	5	无此账号
密码	6	未填写密码
	7	密码过于简单
	8	密码位数过短
	9	密码位数过长
	10	密码和账号不匹配

测试用例表				
用例编号	输入类型		预期输出	覆盖问题
	账号	密码		
1		123456A	拒绝(无效)	1
2	1.11111E+11	123457A	拒绝(无效)	2
3	11	123458A	拒绝(无效)	3
4	#$%	123459A	拒绝(无效)	4
5	123456	123460A	拒绝(无效)	5
6	12345678		拒绝(无效)	6
7	12345678	123456	拒绝(无效)	7
8	12345678	12	拒绝(无效)	8
9	12345678	1.11111E+17	拒绝(无效)	9
10	12345678	12345678	拒绝(无效)	10

3、有一个在线购物网站系统，主要功能包括登录、商品选购、在线支付完成购物等。

用户在使用这些功能时可能会出现各种情况，如账号不存在、密码错误、账户余额不足等。设目前该系统中仅有一个账号 abc；密码为 123；账户余额 200；仅有商品 A，售价均为 50 元，库存为 15，商品 B 售价为 50 元，库存为 0。
试采用场景法：
（1）分析画出事件流图，标注出基本流和备选流；
（2）分析生成测试场景。
（3）对每一个场景设计相应的测试用例。



场景表		
场景1	基本流	
场景2	基本流	备选流1
场景3	基本流	备选流2
场景4	基本流	备选流3
场景5	基本流	备选流4

测试用例表					
用例id	场景/条件	账号	密码	操作	预期结果
1	成功购物	abc	123	登录系统,选购一个有库存,价值为50的货物	支付成功,生成订单
2	账号不存在	zbc	n/a	登录系统	登录失败
3	密码错误	abc	345	登录系统	登录失败
4	货物缺货	abc	123	登录系统,选购一个有库存,价值为50的货物	提示货物无库存,需要重新选购
5	余额不足	abc	123	登录系统,选购一个有库存,价值为500的货物	提示余额不足,购买失败

4、有一个“用户信息输入”界面如下图，输入项有 3 个：姓名、昵称、手机号码，状态有两个：填与不填。请采用正交实验法对其进行测试。
(1）请选择一个合适的正交表：$L_4(2^3)$
(2）根据选定的正交表进行变量映射
(3）写出测试用例

正交表			
	列号		
	1	2	3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

变量映射

映射表				
		列号		
		姓名	昵称	手机号码
行号	1	填	填	填
	2	填	不填	不填
	3	不填	填	不填
	4	不填	不填	填

测试用例表				
用例id	输入状态			预取结果
	姓名	昵称	手机号码	
1	填	填	填	接受(有效)
2	填	不填	不填	拒绝(无效)
3	不填	填	不填	拒绝(无效)
4	不填	不填	填	拒绝(无效)
增补测试用例				
5	不填	不填	不填	拒绝(无效)

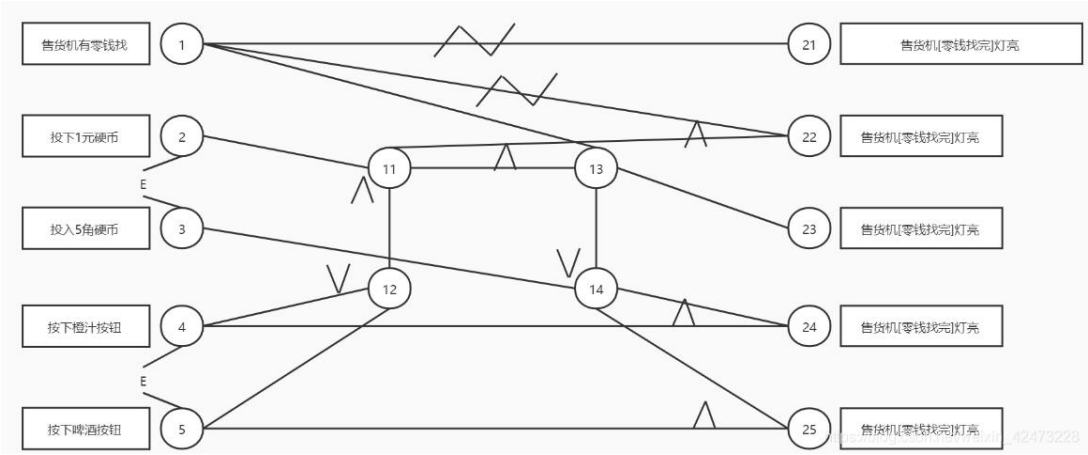
5、有一个处理单价为 5 角钱的饮料自动售货机软件，其规格说明如下：若投入 5 角钱或 1 元钱的硬币，押下〖橙汁〗或〖啤酒〗的按钮，则相应的饮料就送出来。若售货机没有零钱找，则一个显示〖零钱找完〗的红灯亮，这时在投入 1 元硬币并押下按钮后，饮料不送出来而且 1 元硬币也退出来；若有零钱找，则显示〖零钱找完〗的红灯灭，在送出饮料的同时退还 5 角硬币。

(1) 分析软件规格说明，列出原因和结果。

(2) 画出因果图。

(3) 列出简化后的判定表

原因结果表		
序号	原因	结果
1	售货机有零钱找	售货机[零钱找完]灯亮
2	投下1元硬币	退还1元硬币
3	投入5角硬币	退还5角硬币
4	按下橙汁按钮	送出橙汁饮料
5	按下啤酒按钮	送出啤酒饮料



判定表

序号		1	2	3	4	5	6	7	8	9		10	1	2	3	4	5	6	7	8	9		20	1	2	3	4	5	6	7	8	9		30	1	2
条件	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	0	0	0		0	0	0	0	0	0	0	0	0	0		0	0	0
	2	1	1	1	1	1	1	1	1	0		0	0	0	0	0	0	0	1	1	1		1	1	1	1	1	0	0	0	0	0		0	0	0
	3	1	1	1	1	0	0	0	0	1		1	1	1	0	0	0	0	1	1	1		1	0	0	0	0	1	1	1	1	0		0	0	0
	4	1	1	0	0	1	1	0	0	1		1	0	0	1	1	0	0	1	1	0		0	1	1	1	0	1	1	0	0	1		1	0	0
	5	1	0	1	0	1	0	1	0	1		0	1	0	1	0	1	0	1	0	1		0	1	0	0	1	1	0	1	0	1		0	1	0
中间结果	11						1	1	0			0	0	0		0	0	0							1	1	0		0	0	0			0	0	0
	12						1	1	0			1	1	0		1	1	0							1	1	0		1	1	0			1	1	0
	13						1	1	0			0	0	0		0	0	0							0	0	0		0	0	0			0	0	0
	14						1	1	0			1	1	1		0	0	0							0	0	0		1	1	1			0	0	0
结果	21						0	0	0			0	0	0		0	0	0							1	1	1		1	1	1			1	1	1
	22						0	0	0			0	0	0		0	0	0							1	1	0		0	0	0			0	0	0
	23						1	1	0			0	0	0		0	0	0							0	0	0		0	0	0			0	0	0
	24						1	0	0			1	0	0		0	0	0							0	0	0		1	0	0			0	0	0
	25						0	1	0			0	1	0		0	0	0							0	0	0		0	1	0			0	0	0

5、某软件需求规格说明包含如下要求：第一列字符必须是 A 或 B，第二列字符必须是一个数字，在此情况下进行文件修改。但是，如果第一列字符不正确，则输出信息 L；如果第二列字符不是数字，则给出信息 M。请采用因果图进行分析，并绘制出该软件需求规格说明对应的因果图。

6、某程序功能为输出某个输入日期明天的日期，例如输入 2020 年 2 月 2 日，则该程序的输出为 2020 年 2 月 3 日。该程序有三个输入变量 year、month、day，分别表示输入日期的年、月、日。

(1) 请根据程序规格，分别为输入变量 year、month、day 划分有效等价类。

(2) 分析程序的规格说明，并结合以上等价类划分的情况，给出程序所有可能采取的操作。

(3) 根据(1)和(2)，画出简化后的决策表，并为每条规则设计测试用例。

等价类划分表		
输入类型	序号	有效等价类
year	Y1	year 是平年
	Y2	year 是闰年
month	M1	month=2
	M2	month=12
	M3	month=1, 3, 5, 7, 8, 10
	M4	month=4, 6, 9, 11
day	D1	1≤day≤27

	D2	day=28
	D3	day=29
	D4	day=30
	D5	day=31

可能采取操作表	
序号	操作
A1	year+1
A2	month=1
A3	month+1
A4	day=1
A5	day+1
A6	提示输入日期无效

	等价类	1	2	3	4	5	6	7	8	9	10	11	12	13
年	平年				√			√						
	闰年			√			√							
月	2			√	√		√	√	√			√		
	12		√			√				√			√	
	1,3,5,7,8,10		√			√				√				√
	4,6,9,11		√			√					√	√		
日	1-27	√												
	28		√	√	√									
	29					√	√	√						
	30								√	√	√			
	31											√	√	√
动作桩	year+1												√	
	month=1												√	
	month+1				√		√				√			√
	day=1				√		√				√		√	√
	day+1	√	√	√		√				√				
	提示输入日期无效							√	√			√		
测试输入	年	2017	2017	2016	2017	2017	2000	2017	2017	2017	2017	2017	2017	2017
	月	1	12	2	2	3	2	2	2	5	4	9	12	10
	日	1	28	28	28	29	29	29	30	30	30	31	31	31
预期结果	年	2017	2017	2016	2017	2017	2000			2017	2017		2018	
	月	1	12	2	3	3	3			5	5		1	
	日	2	29	29	1	30	1			31	1		1	
	提示输入日期无效							√	√			√		

第四章

一、选择题

1 下列不属于白盒测试的技术是 （ ） 。
A、语句覆盖 B、判定覆盖 C、边界值分析 D、基本路径测试
2 某次程序调试没有出现预计的结果，下列（ ）不可能是导致出错的原因。
A、变量没有初始化
B、编写的语句书写格式不规范
C、循环控制出错
D、代码输入有误
3 代码检查法有桌面检查法，走查和（ ）。
A、静态测试 B、 代码审查 C、动态测试 D、白盒测试
4 如果某测试用例集实现了某软件的路径覆盖，那么它一定同时实现了该软件的 （ ）
A、判定覆盖 B、条件覆盖 C、判定/条件覆盖 D、组合覆盖
5 软件测试的局限性不包括（ ）
A、因为输入/状态空间的无限性，测试不可能完全彻底。
B、巧合性有时会导致错误的代码得到正确的结果，掩盖了问题。
C、软件测试会导致成本增加，效益降低。
D、软件缺陷的不确定性。
6 以下哪种测试方法不属于白盒测试技术（ ）
A、基本路径测试 B、边界值分析测试 C、程序插桩 D、逻辑覆盖测试
7 调试是（ ）
A、发现与预先定义的规格和标准不符合的问题
B、发现软件错误征兆的过程
C、有计划的、可重复的过程
D、消除软件错误的过程
8 使用白盒测试方法时，确定测试数据的依据是指定的覆盖标准和 （ ）
A、程序的注释 B、程序的内部逻辑 C、用户使用说明书 D、程序的需求说明
9 数据流覆盖关注的是程序中某个变量从其声明、赋值到引用的变化情况，它是下列哪一种覆盖的变种（ ）。
A、语句覆盖 B、控制覆盖 C、分支覆盖 D、路径覆盖
10 如果一个判定中的复合条件表达式为（A > 1）or（B <= 3），则为了达到 100%的条件覆盖率，至少需要设计多少个测试用例（ ）。
A、1 B、2 C、3 D、4
11 一个程序中所含有的路径数与（ ）有着直接的关系。
A、程序的复杂程度 B、程序语句行数 C、程序模块数 D、程序指令执行时间
12 条件覆盖的目的是 （ ）
A、使每个判定中的每个条件的可能取值至少满足一次
B、使程序中的每个判定至少都获得一次"真"值和"假"值。
C、使每个判定中的所有条件的所有可能取值组合至少出现一次。
D、使程序中的每个可执行语句至少执行一次。
13 软件调试的目的是（ ）

A、发现软件中隐藏的错误
B、解决测试中发现的错误
C、尽量不发现错误以便早日提交软件
D、证明软件的正确性
14 针对下面一个程序段：
其中，FUCTION1、FUCTION2 均为语句块。现在选取测试用例：M=10 N=0 P=3 ，该测试用例满足了（ ）。

```
If ((M>0) && (N = = 0))
      FUCTION1;
If ((M = = 10)|| (P > 10))
      FUCTION2;
```

A、路径覆盖 B、条件组合覆盖 C、判定覆盖 D、语句覆盖
15 对下面的计算个人所得税程序中，满足判定覆盖的测试用例是( )。

```
if (income<800)       taxrate=0;
else if (income<=1500)       taxrate=0.05;
else if (income<2000)       taxrate=0.08;
else taxrate=0.1;
```

A、income=(799, 1500, 1999, 2000)　　B、income=(799, 1501, 2000, 2001)
C、income=(800, 1500, 2000, 2001)　　D、income=(800, 1499, 2000, 2001)
16 设有一段程序如下：

```
if (a==b    and c==d or e==f)    do    S1
      else    if (p==q    or s==t) do    S2
            else do S3
```

若要达到“判定—条件覆盖”的要求，最少的测试用例数目是（ ）
A、6 B、8 C、3 D、4
17 下列不属于白盒测试中逻辑覆盖标准的是（ ）。
A、语句覆盖 B、条件覆盖 C、分支覆盖 D、边界值覆盖
18 在某学校的综合管理系统设计阶段,教师实体在学籍管理子系统中被称为"教师",而在人事管理子系统中被称为"职工",这类冲突描述正确的为（ ）。
A、语义冲突 B、命名冲突 C、属性冲突 D、结构冲突

二、填空题

1、代码检查的方式有三种：桌面检查、代码审查、_代码走查__。
2、数据流分析就是对程序中数据的__定义_、__使用_及其之间的_依赖关系_等进行分析的过程。
3、_条件组合覆盖__是逻辑覆盖标准的一种，它要求选取足够多的测试数据，使得每个判定表达式中条件的各种可能组合都至少出现一次。

三、判断题

1、所有满足条件组合覆盖标准的测试用例集，也分支覆盖标准。 （√ ）
2、软件测试的目的在于发现错误、改正错误。 （× ）
3、条件覆盖能够查出条件中包含的错误，但有时达不到判定覆盖的覆盖率要求。 （ √）
4、在白盒测试中，如果某种覆盖率达到 100% ，就可以保证把所有隐藏的程序缺陷都已经揭露出来了。 （ ×）

5、白盒测试的条件覆盖标准强于判定覆盖。 （× ）

6、判定覆盖包含了语句覆盖，但它不能保证每个错误条件都能检查出来。 （ √）

四、解答题

1、请为以下程序段设计测试用例集，要求分别满足语句覆盖、判定覆盖、条件覆盖、条件/判定覆盖覆盖、条件组合覆盖。

```
public int do_work(int A,int B){
        int x=0;
        if((A>4) && (B<9))
          { x = A-B;}
        if( A==5 && B>28 )
          { x= A+B;}
          return x;
    }
```

语句覆盖：			
测试用例编号	测试用例	（A>4）&&(B<9)	A==5&&B>28
1	A=5, B=3	T	
2	A=5, B=30		T

判定覆盖：			
测试用例编号	测试用例	（A>4）&&(B<9)	A==5&&B>28
3	A=5, B=3	T	F
4	A=5, B=30	F	T

条件覆盖：					
测试用例编号	测试用例	(A>4)	(B<9)	A==5	B>28
5	A=5, B=3	T	T	T	F
6	A=3, B=30	F	F	F	T

条件/判定覆盖：							
测试用例编号	测试用例	(A>4)	(B<9)	A==5	B>28	(A>4)&&(B<9)	A==5&&B>28
7	A=5, B=3	T	T	T	F	T	F
8	A=3, B=30	F	F	F	T	F	F
9	A=5, B=30						T

条件组合覆盖：			
测试用例编号	测试用例	（A>4）&&(B<9)	A==5&&B>28
10	A=5, B=3	T、T	T、F
11	A=5, B=30	T、F	T、T
12	A=3, B=3	F、T	F、F
13	A=3, B=30	F、F	F、T

2、为以下程序段设计测试用例集，要求分别满足语句覆盖、判定覆盖、条件覆盖、修正条件/判定覆盖。

```
public int do_work(int x,int y,int z){
        int k=0,int j=0;
        if((x>20) && (z<10))
          { k = x*y-1;
            j=k*k;
          }
        if( (x==22) ||( y>20) )
```

```
            { j= x*y+10;}
            j=j%3;
            System.out.println("k,j is:"+k+"."+j)
          }
```

3、请为以下程序段设计测试用例集，要求满足条件组合覆盖

```
public class Triangle {
    protected long lborderA = 0;
    protected long lborderB = 0;
    protected long lborderC = 0;

    // Constructor
    public Triangle(long lborderA, long lborderB, long lborderC) {
        this.lborderA = lborderA;
        this.lborderB = lborderB;
        this.lborderC = lborderC;
    }

    public boolean isTriangle(Triangle triangle) {
        boolean isTriangle = false;

        // check boundary
        if (triangle.lborderA > 0 && triangle.lborderB > 0 && triangle.lborderC > 0 )
           // check if subtraction of two border larger than the third
        if ((triangle.lborderA-triangle.lborderB) < triangle.lborderC
                    && (triangle.lborderB-triangle.lborderC) < triangle.lborderA
                        && (triangle.lborderC-triangle.lborderA) < triangle.lborderB)
        {isTriangle = true; }
        return isTriangle;
    }
}
```

测试用例 lborderA lborderB lborderC	lborderA>0&& lborderB>0&& lborderC>0	lborderA- lborderB< lborderC	lborderB- lborderC< lborderA	lborderC- lborderA< lborderB
0、0、0	F、F、F			
0、0、1	F、F、T			
0、1、0	F、T、F			
0、1、1	F、T、T			
1、0、0	T、F、F			
1、0、1	T、F、T			
1、1、0	T、T、F			
1、1、1	T、T、T	T	T	T
4、1、2		F		
1、4、2			F	
1、2、4				F

4、请为程序模块 Function1

（1）画出程序控制流图，计算控制流图的环路复杂度

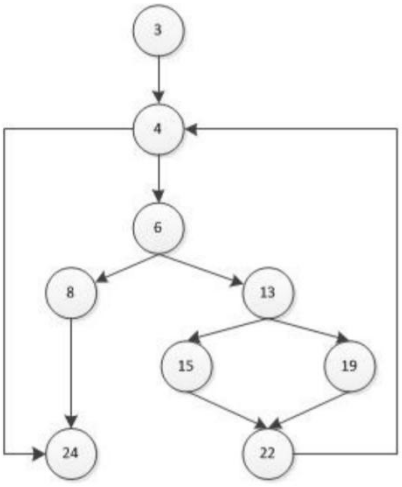
（2）导出基本路径

（3）设计基本路径覆盖测试用例

程序模块 Function1 代码如下：

```
1       public   int   Function1(int num, int cycle, boolean flag)
2       {
3               int ret = 0;
4               while( cycle > 0 )
5               {
6                       if( flag == true )
7                       {
8                               ret = num - 10;
9                               break;
10                      }
11                      else
12                      {
13                              if( num%2 ==0 )
14                              {
15                                      ret = ret * 10;
16                              }
17                              else
18                              {
19                                      ret = ret + 1;
20                              }
21                      }
22                      cycle--;
23              }
24              return ret;
25      }
```

（1）画出程序控制流图



计算控制流图环路复杂度
V(G)=4(图中有四个区域)
（2）导出基本路径
A.3-4-24
B.3-4-6-8-24
C.3-4-6-13-15-22-4-24
D.3-4-6-13-19-22-4-24
（3）设计基本路径覆盖测试用例

测试用例	覆盖基本路径
（10,0，true）	A
（100,10，true）	B
（10,1，false）	C
（5,1，false）	D

5、请对以下程序进行插桩，显示循环执行的次数

```
public class GCD {
	public int getGCD(int x,int y){
		if(x<1||x>100)
		{
			System.out.println("参数不正确!");
			return -1;
		}

		if(y<1||y>100)
		{
			System.out.println("参数不正确!");
			return -1;
		}

		int max,min,result = 1;
		if(x>=y)
		{
			max = x;
			min = y;
		}
		else
		{
			max = y;
			min = x;
		}

		for(int n=1;n<=min;n++)
		{
			if(min%n==0&&max%n==0)
```

```
                {
                    if(n>result)
                    result = n;
                }
            }

            System.out.println("最大公约数为:"+result);
            return result;
        }
 }
```

答案：

```
        int cycle_time = 0;     //插桩
        for(int n=1;n<=min;n++)
        {   cycle_time = cycle_time +1;   //插桩
           if(min%n==0&&max%n==0)
         {     if(n>result)
                result = n;
           }
        }
      System.out.println(“最大公约数为：”+result);
      System.out.println(“循环执行次数为：”+cycle_time);   //插桩
      return result;
}
}
```

6、请对以下代码段进行变异，变异规则为将“++”替换为“–”，然后设计测试数据，能够测试发现所有的变异点。

```
public class zhengchu {
          public     String    iszhengchu(int n) {
                if(n<0||n>500) {
                     return "error";
                }
                int flag=0;
                String note="";
                if(n%3==0) {
                flag++;
                note=note+" 3";
            }

            if(n%5==0) {
                flag++;
                note+=" 5";
            }

            if(n%7==0) {
```

```
            flag++;
            note+=" 7";
        }

        return "能被"+flag+"个数整除,"+note;
    }
}
```

变异代码：

```
public class zhengchu {
        public    String   iszhengchu(int n) {
            if(n<0||n>500) {
                return "error";
            }
            int flag=0;
            String note="";
            if(n%3==0) {
            flag--;
            note=note+" 3";
        }

        if(n%5==0) {
            flag--;
            note+=" 5";
        }

        if(n%7==0) {
            flag--;
            note+=" 7";
        }

        return "能被"+flag+"个数整除,"+note;
    }
}
```

变异测试用例					
测试用例序号	输入类型	输出		发现变异点	备注
	n	预期	实际		
1	3	能被1个数整除 3	能被-1个数整除 3	1	有效
2	5	能被1个数整除 5	能被-1个数整除 5	1	有效
3	7	能被1个数整除 7	能被-1个数整除 7	1	有效
4	15	能被2个数整除 3 5	能被-2个数整除 3 5	2	有效
5	21	能被2个数整除 3 7	能被-2个数整除 3 7	2	有效
6	35	能被2个数整除 5 7	能被-2个数整除 5 7	2	有效
7	105	能被3个数整除 3 5 7	能被-3个数整除 3 5 7	3	有效
8	0	能被0个数整除	能被0个数整除	0	无效

7、试比较调试跟测试的不同

第五章

一、选择题

1 软件测试是软件质量保证的重要手段，下述哪种测试是软件测试的最基础环节？（ ）
A．集成测试 B．单元测试 C．系统测试 D．验收测试
2 增量式集成测试有 3 种方式：自顶向下增量测试方法，（ ）和混合增量测试方式。
A．自下向顶增量测试方法
B．自底向上增量测试方法
C．自顶向上增量测试方法
D．自下向顶增量测试方法
3 在软件测试步骤按次序可以划分为以下几步：（ ）。
A、单元测试、集成测试、系统测试、验收测试
B、验收测试、单元测试、系统测试、集成测试
C、单元测试、集成测试、验收测试、系统测试
D、系统测试、单元测试、集成测试、验收测试
4 软件验收测试合格通过的标准不包括（ ）
A．软件需求分析说明书中定义的所有功能已全部实现，性能指标全部达到要求。
B．至少有一项软件功能超出软件需求分析说明书中的定义，属于软件特色功能。
C．立项审批表、需求分析文档、设计文档和编码实现一致。
D．所有在软件测试中被发现的严重软件缺陷均已被修复。
5 下列关于 alpha 测试的描述中正确的是：（ ）
A．alpha 测试一定要真实的最终软件用户参加
B．alpha 测试是集成测试的一种
C．alpha 测试是系统测试的一种
D．alpha 测试是验收测试的一种
6 编码阶段产生的错误主要由（ ）检查出来的。
A、单元测试 B、集成测试 C、系统测试 D、有效性测试
7 单元测试一般以（ ）为主。
A、白盒测试 B、黑盒测试 C、系统测试 D、分析测试
8 单元测试的测试用例主要根据（ ）的结果来设计。
A、需求分析 B、源程序 C、概要设计 D、详细设计
9 集成测试的测试用例是根据（ ）的结果来设计。
A、需求分析 B、源程序 C、概要设计 D、详细设计
10 集成测试对系统内部的交互以及集成后系统功能检验了何种质量特性( )
A、正确性 B、可靠性 C、安全性 D、可维护性
11（ ）的目的是对即将交付使用的软件系统进行全面的测试，确保最终软件产品满足用户需求。
A、系统测试 B、集成测试 C、单元测试 D、验收测试
12 单元测试中用来模拟被测模块调用者的模块是 （ ）
A、父模块 B、子模块 C、驱动模块 D、桩模块
13 在自底向上测试中，要编写（ ）。
A、测试存根 B、驱动模块 C、桩模块 D、 底层模块。
14 以下哪种软件测试属于软件性能测试的范畴（ ）。

A、接口测试 B、压力测试 C、单元测试 D、正确性测试
15 下列关于α测试的描述中，正确的是（ ）
A．α测试采用白盒测试技术；
B．α测试不需要从用户角度考虑问题；
C．α测试是系统测试的一种；
D．α测试是验收测试的一种；
16 下列软件属性中，软件产品首要满足的应该是（ ）
A、功能需求 B、性能需求 C、可扩展性和灵活性 D、容错纠错能力
17 按照测试组织划分，软件测试可分为：开发方测试，第三方测试，()。
Ａ．集成测试 Ｂ．单元测试 Ｃ．用户测试 Ｄ．灰盒测试
18 软件可靠性是指在指定的条件下使用时，软件产品维持规定的性能级别的能力，其子特性（ ）是指在软件发生故障或者违反指定接口的情况下，软件产品维持规定的性能级别的能力。
A、成熟性 B、易恢复性 C、容错性 D、稳定性
19 下面哪项对验收测试的描述不正确？（ ）
A、与系统测试不同的是以客户业务需求为标准来进行测试
B、测试人员多由客户方担任，也可以客户委托第三方来进行验收测试
C、由资深的开发和测试人员来进行测试
D、不仅仅要验收程序，还要验收相关的文档
20 对于软件的β测试，下列哪些描述是正确的？ （ ）
A. β测试就是在软件公司内部展开的测试，由公司专业的测试人员执行的测试。
B. β测试就是在软件公司内部展开的测试，由公司的非专业测试人员执行的测试。
C. β测试就是在软件公司外部展开的测试，由非专业的测试人员执行的测试。
D. β测试就是在软件公司外部展开的测试，由专业的测试人员执行的测试。
21 在程序测试中，用于检查程序模块或子程序之间的调用是否正确的静态分析方法是 （ ）
A、操作性分析 B、可靠性分析 C、引用分析 D、接口分析
22 用于考察当前软硬件环境下软件系统所能承受的最大负荷并帮助找出系统瓶颈所在的是（ ）。
A、压力测试 B、容量测试 C、负载测试 D、疲劳测试

二、填空题

1、集成测试以_概要设计_说明书为指导，验收测试以_软件需求__说明书为指导。
2、软件验收测试可分为 2 类：__α+β测试__、_正式验收测试__。
3、_回归测试__指软件系统被修改或扩充后重新进行的测试。
4、α测试___是在软件开发公司内模拟软件系统的运行环境下的一种验收测试。
5、_系统测试___的依据是软件规格说明书。

三、判断题

1、单元测试通常由开发人员进行。 （√ ）
2、测试应从"大规模"开始，逐步转向"小规模"。 （× ）
3、负载测试是验证要检验的系统的能力最高能达到什么程度。 （× ）
4、为了快速完成集成测试，采用一次性集成方式是最适宜的（×）
5、验收测试是站在用户角度的测试。 （√ ）
6、自底向上集成需要测试员编写桩模块。 （× ）
7、β测试是集成测试的一种。 （× ）
8、如何看待软件产品内部的缺陷，开发者和用户的立场是一致的。 （× ）

四、解答题

1、试针对如下程序代码设计测试脚本。

```
public class GCD {
	public int getGCD(int x,int y) {
		if(x<1||x>100) {
			System.out.println("数据超出范围!");
			return -1;
		}
		if(y<1||y>100) {
			System.out.println("数据超出范围!");
			return -1;
		}
		int max,min,result = 1;
		if(x>=y) {
			max = x;
			min = y;
		}
		else {
			max = y;
			min = x;
		}
		for(int n=1;n<=min;n++) {
			if(min%n==0&&max%n==0) {
				if(n>result)
					result = n;
			}
		}
		System.out.println("因数:"+result);
		return result;
	}
}
```

1）设计测试脚本，对 GCD 类的 getGCD 方法实现语句覆盖测试。

```
public class GCD {
	public void testDemo_yuju(){
		int test1=getGCD(-1,100);
		int test2=getGCD(2,1);
		System.println.out(test1+ test2);
	}
	public int getGCD(int x,int y) {
		if(x<1||x>100) {
			System.out.println("数据超出范围!");
			return -1;
		}
		if(y<1||y>100) {
```

```
            System.out.println("数据超出范围!");
            return -1;
        }
        int max,min,result = 1;
        if(x>=y) {
            max = x;
            min = y;
        }
        else {
            max = y;
            min = x;
        }
        for(int n=1;n<=min;n++) {
            if(min%n==0&&max%n==0) {
                    if(n>result)
                    result = n;
            }
        }
        System.out.println("因数:"+result);
        return result;
    }
}
```

（2）设计测试脚本，对 GCD 类的 getGCD 方法实现条件覆盖测试。

```
public class GCD {
    public void testDemo_tiaojian(){
        int test1=getGCD(-1,-1);
        int test2=getGCD(100,100);
        int test3=getGCD(2,1);
        int test4=getGCD(1,2);
        System.println.out(test1+ test2+ test3+ test4);
    }
    public int getGCD(int x,int y) {
         if(x<1||x>100) {
                System.out.println("数据超出范围!");
                return -1;
            }
            if(y<1||y>100) {
                System.out.println("数据超出范围!");
                return -1;
            }
            int max,min,result = 1;
            if(x>=y) {
                max = x;
                min = y;
```

```
                }
                else {
                        max = y;
                        min = x;
                }
                for(int n=1;n<=min;n++) {
                        if(min%n==0&&max%n==0) {
                                if(n>result)
                                result = n;
                        }
                }
                System.out.println("因数:"+result);
                return result;
        }
}
```

2、设有程序段 ModuleA 和 ModuleB 如下，

```
public class ModuleA {
        public static double operate(double x) {
                //  模块 A 内部进行处理
                // ...
                double r = x/2;
                //  调用模块 B
                double y = ModuleB.operate(r);
                //  继续处理
                // ...
                return y;
        }
}
public class ModuleB {
        public static double operate(double r) {
                //  模块 B 内部进行处理
                // ...
                double temp = Pi*r * r * r *4/3;
                //  继续处理
                // ...
                double y = temp;
                return y;
        }
}
```

（1）阅读程序，请说明这两段程序合起来的功能是什么？

输入直径,求圆球的体积

（2）已知变量 x 一开始就有一定的误差△x，请分析 ModuleA.operate(x)执行完毕后，返回结果 y 的相对误差有多大？

相对误差分析：$y=Pi*(x/2)^3*4/3$
两边微分得 $dy=Pi/6*3x^2dx$,两边再同除以 y 和 $Pi*(x/2)^3*4/3$ 得：

$$dy/y = 3\ dx/x$$

3、设有两段代码 ModuleA 和 ModuleB 如下，它们由不同的程序员开发。

```
public class ModuleA {
    /*** 实现把 str1 中包含的 str2 去掉后的内容返回的功能
    * @param str1 字符串 1
    * @param str2 字符串 2
    * @param 返回处理的结果
    */
    public String operate(String str1, String str2) {
        return str1.replace(str2, "");
    }
}

public class ModuleB {
    private ModuleA moduleA;
    public void setModuleA(ModuleA moduleA) {
        this.moduleA = moduleA;
    }
    /**
    * 模块 B 的具体处理操作中，调用了模块 A 的接口
    */
    public String operate(String str1, String str2) {
        // str1 待替换的目标串
        // str2 原串
        return moduleA.operate(str1, str2);
    }
}
```

（1）试分析对这两段代码进行集成测试时会出现什么问题？

ModuleB 调用 ModuleA 时给的参数次序有误，会出错。

（2）试设计两个测试数据，一个能发现这一问题，另一个则不能发现这一问题。

能发现这一问题的测试用例：str1=“B”，str2=“AB”

不能发现这一问题的测试用例：str1=“ABC”，str2=“ABC”

3、试分析集成测试与系统测试的区别。

（1）测试对象不同。集成测试的测试对象只是软件本身，而系统测试的测试对象是包括软件及其运行环境组成的整个系统。

（2）测试依据不同。集成测试的依据是系统概要设计，而系统测试的依据是软件规格说明书。

（3）测试方法不同。集成测试通常采用白盒测试加黑盒测试的方法，而系统测试完全采用黑盒测试方法。

（4）测试内容侧重点不同。集成测试的侧重点是各个单元模块之间的接口，以及各个模块集成后所实现的功能；而系统测试的主要内容就是整个系统的功能、性能、安全性等。

（5）测试人员不同。集成测试工作一般由开发人员或者开发人员和测试人员共同完成，而系统测试一般由专门的测试人员完成。
4、某连锁机构网站有注册账号 5 万个，平均 1 天大约有 12000 个用户要访问该系统，用户一般在 7 点——22 点使用该系统，在一天的时间内，用户使用系统的平均时长约为 0.5 小时。假设用户登录访问该系统符合泊松分布，为进行并发测试，请估算系统的平均并发用户数 C_avg 和并发用户峰值数 C_max。

系统的平均并发用户数 C_avg=nL/T=12000*0.5/(22-7)=400
并发用户峰值数 C_max~=400+3*400^1/2=460

第六章

一、选择题

1 面向对象软件测试不包括（ ）。
A、分析与设计模型测试技术 B、类的封装技术
C、类测试技术 D、对象交互测试技术
2 以下表述中错误的是（ ）。
A、类测试工作的时间一般在完全说明一个类，并且准备对其编码后。
B、应该在软件的其它部分使用该类之后来执行对类的测试。
C、在测试类的功能实现时，应该首先保证类成员函数的正确性。
D、应防止因未经测试的类被使用而导致缺陷传导和扩散。

二、填空题

1、封装这一特征简化了对对象的使用，但同时也给_测试结构的分析_、_测试路径的选取_、_测试数据的生成__等带来了困难。
2、通过继承机制，子类可以继承父类的特点和功能，这一特征为_缺陷的扩散_提供了途径
3、__类测试_是由那些与验证类的实现是否和该类的说明完全一致的相关联的活动组成的。
4、类测试的目的是要确保一个类的代码能够完全满足__类的说明__所描述的要求。
5、每当一个类的实现发生变化时，就应该执行_回归测试___。

三、判断题

1、虽然类的实现正确，但类的每一个实例的行为不一定是正确的。（ ×）
2、软件测试等于程序测试。（ ×）
3、设计－实现－测试，软件测试是开发后期的一个阶段。（× ）
4、类测试应测试到操作和状态转换的所有组合情况。（× ）

四、解答题

什么是多态，多态对测试有什么影响？
多态是同一个操作作用于不同的对象，可以有不同的解释，产生不同的执行结果。
多态性增强了软件的灵活性和重用性，同时也使得测试的工作量成倍增加。
多态和动态绑定增加了软件运行中可能的执行路径，而且给面向对象软件带来了不确定性，给测试覆盖带来了困难。

第七章

一、选择题

1 下列（）不是软件自动化测试的优点

A. 速度快、效率高　　B.准确度和精确度高

C.能节约测试工作的人力成本　　D.能完全代替手工测试工作

2 关于自动化测试局限性的描述，以下描述错误的是（）

A. 自动化测试不能完全取代手工测试

B. 自动测试比手工测试发现的缺陷少

C. 自动化测试不能提高测试的覆盖率

D. 自动化测试对测试设计依赖性极大

3 以下不适用自动化测试的情况的是（）

A. 负载测试　　B.回归测试　　C.界面体验测试　　D.压力测试

二、填空题

1、自动化测试中，实现对中间结果进行检查的技术是数据验证点，实现重复执行脚本并每次输入不同测试用例的技术是数据驱动。

2、自动化测试的基本原理大致分为两类：一是通过设计的特殊程序模拟测试人员对计算机的操作过程和操作行为，一般用来实现自动化黑盒测试；二是开发类似于高级编译系统那样的软件分析系统，来对被测试程序进行检查、分析和质量度量等，一般用来实现自动化白盒测试。

3、自动化测试只是提高了测试执行的效率，而不能提高测试的有效性。

三、判断题

1、只要采用自动化测试，工作效率将马上提高。（×）

2、所有的测试工作都可以实现自动化。（×）

3、自动化测试的执行是不受上下班时间限制的，甚至于可以 24 小时不间断。（√）

4、在相同的测试设计、执行相同的测试数据的情况下，自动化测试比手工测试发现的缺陷多。（×）

5、测试用例可完全由测试工具自动生成。（×）

6、自动化测试可适用于任何测试场景。（×）

四、解答题

1、试分析应用自动化测试技术应注意哪些问题。

自动化测试有它的优点，也有局限性。

（1）自动化测试并不比手工测试发现的缺陷更多。自动化测试主要是把测试的执行过程交给了计算机来自动完成，而能发现多少缺陷主要是测试设计决定的。简单地说，在相同的测试设计、执行相同的测试数据的情况下，自动化执行和手工执行测试发现的缺陷是一样多的。自动化测试只是提高了测试执行的效率，并不能提高测试的有效性。

（2）自动化测试脚本或程序自身也需要进行正确性检查和验证。自动化测试脚本或程序也是由人开发出来的，也存在出错的可能性，因而也需要对其进行正确性检查和验证。

（3）自动化测试对测试设计的依赖性很大。自动化测试要能够顺利执行并达到测试目的，它对测试设计的依赖性很大，要事先设计测试规程、测试数据、搭建测试环境，测试设计的质量更为关键，自动化测试工具本身只是起到辅助作用。

（4）自动化测试比手工测试更加“脆弱”，并需要进行维护。自动化测试有非常具体的执行条件，

执行过程也是固定的，当被测试程序有修改或者测试环境条件有变化时，可能就无法执行，非常“脆弱”。为适应程序的修改、扩充，或者是环境条件的变化，自动化测试脚本和代码需要不断进行维护。

（5）自动化测试也需要相应的成本投入。实现自动化测试需要进行测试人员培训、测试工具购买、测试环境部署、测试脚本或程序开发等，也会有相应的成本投入，尤其是初期，比手工测试的开销更大。

2、试分析以下测试脚本每一行代码的功能是什么。

```
startApp(“校园招聘”);    //启动应用软件 校园招聘
tree().click(atPath(“学校->专业->班级”));    //在显示的目录树中依次选择 学校、专业、班级
...
学号().inputKeys(“{Num1}{Num8}{Num1}{Num2}{Num3}{Num4}{Num1}{Num2}{Num3}{Num4}”); //输入学号“1812341234”
查询().click();    // 单击“查询”按钮
校园招聘(ANY,MAY_EXIT).close();    //关闭应用软件 校园招聘
```

3、试结合实例阐述数据验证点在使用时可以达到哪些不同的效果。

（1）借助于在脚本中插入数据验证点，可以在脚本回放时进行数据检查验证，以判断测试过程或结果是否正确。例如，以下脚本代码行用于插入数据验证点检验被测软件计算得到的总金额是否等于预定值：_15090().performTest(OrderTotalAmountVP());

（2）数据验证点除了可以判断测试过程或结果是否正确之外，还可以实现脚本代码执行和界面显示之间的同步。例如，测试流程为：在前一个界面执行后，弹出后一个界面，然后在后一个界面单击“OK”按钮。但可能当脚本代码执行到要在后一个界面单击“OK”按钮时，后一个界面“OK”按钮还没有显示出来，此时，应在实现单击“OK”按钮的代码行之前，插入数据验证点，检查后一个界面“OK”按钮是否已经显示出来。

第八章

一、选择题

1 对软件文档的要求不包括（ ）。

A、完整性 B、美观性 C、一致性 D、易理解性。

2 软件设计阶段的质量控制主要采取的方式是（ ）。

A、评审 B、白盒测试 C、黑盒测试 D、动态测试

3 以下不属于软件评审内容的是（ ）。

A、管理评审 B、技术评审 C、文档评审 D、人员评审

4 以下不是评审工具的是（ ）。

A、Gerrit B、Jupiter C、JaCoCo D、SourceMonitor

二、填空题

1、评审会议结束后，应当整理得到_评审报告__作为存档材料。

2、对评审会议发现的问题和缺陷要进行分析和跟踪，有的缺陷将_被有条件的接受__，有的缺陷则必须_修正__。

3、验收评审的内容主要是：开发的软件系统是否已达到_软件需求说明书____规定的各项技术指标；_使用手册__是否完整、正确；_文档__是否齐全，是否符合有关标准等。

4、按照 IEEE 的定义，__软件评审__是软件开发组之外的人员或小组，对软件需求、设计或代码，进行详细检查的一种正式评价方法。

5、除软件测试之外，_软件评审___是另一种软件质量控制和软件质量保证的有效方法。
6、大中型软件的质量更多的取决于_需求分析__和__软件设计_质量，而不仅仅是编码质量。
7、正式评审一般以__评审会___的形式进行。

三、判断题

1、技术评审既是一种技术手段，也是一种质量管理手段。 （√ ）
2、详细设计评审是所有的评审活动中最难的一个。 （× ）
3、评审的主要目标在于检测错误、核对与标准的偏离。 （√ ）
4、数据库设计一般要求遵循 4NF。 （× ）
5、应选择那些最复杂和最危险的部分进行评审。 （√ ）
6、应该将发现缺陷的工作推后，最后来处理，这样效率高。 （ ×）

四、解答题

1、试分析通过评审可以有哪些收效？
（1）及时发现软件开发过程中可能引入的缺陷，保证质量。在整个软件开发过程中，每一个阶段都有可能引入缺陷。每一个阶段性软件产品，尤其是不便进行软件测试的阶段性产品，通过对其评审，可以及时发现可能引入的缺陷，保证质量，防止缺陷传导到后续的环节。
（2）提高软件生产率。这是由于通过评审尽早发现了软件缺陷，修复缺陷的工作量小，时间短，并可以减少后续可能需要返工的时间，还可能减少软件测试的时间。反之，则随着开发工作的推进，早期引入的缺陷也必然相应的放大和扩散，由于没有及时排除，在后期将更难于发现，也难于修复，甚至需要推翻重来，造成软件项目生产率低下。
（3）降低软件质量成本。越早发现问题，解决问题，成本越低。通过评审及时发现软件缺陷，可以降低修复缺陷的成本。通过评审可以保证软件阶段性产品的质量，降低后续的测试成本，并降低软件失效的概率，从而从整体上降低软件质量成本。
（4）实现项目监控，标志阶段完成。通过在各个阶段进行评审，可以及时获得客观、可信的项目情况信息，实现对项目进展的监控。同时每次评审通过的软件元素都是后续工作的基线，标志着一个阶段的完成，可以进入下一个阶段。
（5）发现通过测试无法实现的缺陷。例如，评审可以发现下列通过测试无法发现的问题：对标准的符合、风格的一致性、逻辑性问题、模块化问题等。
（6）通过评审学习和积累。参加评审的人员可以通过评审互相交流和学习，从评审发现的问题中吸取有益的教训，通过分析问题和缺陷的原因，可以学到知识，并使人们在以后的工作和项目中避免出现类似的问题。可以将评审发现的典型缺陷和问题列成检查表，用于以后的评审工作等。
2、什么是软件评审，主要的分阶段软件评审活动有哪些？
软件评审是软件软件开发组之外的人员或小组，对软件需求、设计或代码，进行详细审查的一种正式评价方法。
其目的是要发现软件中的缺陷，找出违背执行标准的情况以及其他问题。
不同的软件开发组织和软件项目实施评审的做法有所不同，主要的软件评审活动有：软件项目可行性评审、需求评审、设计评审、代码评审、测试评审等等。

第九章

一、选择题

1 软件质量保证与测试人员需要的的基本素质有（ ）

A、计算机专业技能 B、测试专业技能 C、行业知识 D、以上都是

2CMM 中文全称为（ ）

A、软件能力成熟度模型

B、软件能力成熟度模型集成

C、质量管理体系

D、软件工程研究所

3CMM 将软件组织的软件能力成熟度描述为（ ）

A、二级 B、三级 C、四级 D、五级

4 软件的六大质量特性包括（ ）。

①功能性、可靠性 ②可用性、效率 ③稳定性、可移植 ④多语言性、可扩展性

A、①②③ B、②③④ C、①③④ D、①②④

5 软件验证和确认是保证软件质量的重要措施，它的实施应该针对（ ）

A、程序编写阶段 B、软件开发的所有阶段

C、软件调试阶段 D、软件设计阶段

二、填空题

1、软件对属于各种质量因素的需求的符合性是由_软件质量度量__来测量的。

2、Burnstein 博士提出了 Test Maturity Model(TMM 即软件能力成熟度模型)，它描述了测试过程，是软件测试得到良好计划和控制的基础。

3、按照时间点来区分，软件质量特性度量有两类预测型_和_验收型。

4、CMM 内容包含初始级、_已定义级_、_已管理级、可重复级和可优化级五个等级。

5、McCall 模型划分了_产品修正_、_产品转移_、_产品运行_三个纬度的 11 个软件质量因素。

6、_软件质量_是指软件产品中能满足给定需求的性质和特性的总体。

三、判断题

1、软件质量保证的独特性是由软件产品不同于其他制造产品的本质决定的。（√ ）

2、TMM 分解为 3 个级别，在最高级中，测试不是行为，而是一种自觉的约束，不用太多的测试投入，即可产生低风险的软件。（× ）

3、CMMI 并不包括 CMM，更加适用于企业的过程改进实施。（× ）

4、只有客户才会有兴趣透彻定义软件需求以确保他约定的软件产品的质量。（× ）

四、解答题

1、某软件公司为某电影院设计开发了一款票务系统，包括票务管理、账号管理、在线购票、统计分析等功能，该软件计划长期使用，部分模块将用于其他类似软件，软件在使用时应能接入数字化城市平台。试结合软件质量模型分析应从哪些特性来分析评价这一软件的质量。

可采用 McCall 软件质量模型，从 3 个方面，11 个特性分析这一软件的质量，

例如，
（1）软件各项功能应当正确实现
（2）在线购票等高并发模块应具有较高的执行效率，保证在有很多人购票的情况下，系统能有较好的性能
（3）该系统计划长期使用，应当有较好的可维护性
（4）部分模块将用于其他类型软件，这些模块应有较好的可移植性、可复用性
（5）软件在使用时需接入数字化城市平台，应当具有较好的互联性
2、试分析 SQA 活动主要包含哪些，并举例说明。
（1）识别软件质量需求，并将其自顶向下逐步分解为可以度量和控制的质量要素，为软件开发、维护各阶段软件质量的定性分析和定量度量打下基础。
（2）参与软件项目计划制订。
（3）制订 SQA 计划。
（4）评审工作产品。
（5）审核项目活动。
（6）生成 SQA 报告。
（7）处理不合格项，跟踪问题。
（8）监控软件过程和产品质量。

第十章

一、选择题

1 以下哪句话是不正确的：
A、测试过的软件就没有缺陷
B、测试的目的是尽可能多的发现程序中的缺陷
C、成功的测试在于发现了迄今尚未发现的缺陷
D、测试是为了验证程序是否符合需求
2 下列项目中不属于测试文档的是 （ ）
A、测试计划 B、测试用例 C、 被测程序 D、测试报告
3 软件测试管理不包括： （ ）
A、测试团队管理 B、缺陷管理 C、软件需求管理 D、测试用例管理
4 软件测试风险管理包含（ ）和风险控制两方面内容。
A、风险排序 B、风险识别 C、风险评估 D、风险分析
5 编写测试计划的目的不包括（ ）
A、使测试工作顺利进行
B、使项目参与人员沟通更舒畅
C、使测试工作更加系统化
D、使测试内容更少，完成更快
6 下面哪项内容不包含在测试计划文档中？（ ）
A. 测试策略 B. 测试用例 C. 测试时间安排 D. 测试标准

二、填空题

1、软件测试项目的生命周期包括测试需求分析、__测试计划_、_测试设计__、测试开发、测试执行、评估（或总结）等阶段。
2、软件测试中，_测试计划_描述测试的整体方案，_缺陷报告_描述依据测试用例找出的问题。

3、_软件测试项目管理_就是以测试项目为管理对象，通过一个临时性的专门的测试组织，运用专门的软件测试知识、技能、工具和方法，对测试项目进行计划、组织、执行和控制，并在时间成本、软件测试质量等方面进行分析和管理活动。
4、软件测试文档为测试项目的组织、规划和管理提供了一个_规范化的架构__。

三、判断题

1、测试人员要坚持原则，缺陷未完全修复坚决不予通过。 （× ）
2、在软件测试中, 预设输出结果是检验待测系统在特定执行下是否正确的方法。 （ √ ）
3、发现缺陷越多的模块隐藏的缺陷可能也越少。 （× ）

四、解答题

1、什么是软件测试文档，测试项目中，主要的测试文档有哪些？
软件测试文档是对要执行的测试及测试的结果进行描述、定义、规定和报告的任何书面或图示信息。它为测试项目的组织、规划和管理提供了一个规范化的架构。
主要的测试文档有:测试需求分析、测试计划书、测试设计书、测试用例说明、测试规程规格说明、测试日志、测试执行记录、测试缺陷报告、测试总结报告等。
2、软件测试工作和软件开发工作相比，有哪些特点。
软件测试工作和软件开发工作相比，有其自身的特点。
最显著的特点是，软件测试不直接生产软件产品，较为难以直观体现工作者的工作价值。实际上软件测试的工作价值体现在降低软件失败的风险进而降低软件失败成本。
其次经验在软件测试中很重要，包括软件开发经验和软件测试经验，没有经验的测试人员可能执行了大量测试却发现不了问题，而有经验的测试人员却可能一下子就能发现软件中的问题。
第三，有些测试工作可能没有没有先例可循，需要测试人员根据实际情况创造性的设计测试方案、执行测试过程、得出可靠的测试结论。而且，即使是完成同样的测试任务，不同的测试人员设计的测试方案、测试用例可能在测试成本和测试效果上差距很大，也就是说测试人员的创造性劳动对测试工作很重要。
第四，软件测试是要去发现软件中的问题，有时这些问题并不容易暴露出来，因此细心和耐心对测试人员很重要。
第五，一方面测试工作中有一些简单重复劳动，另一方面随着技术的发展，一些测试工作的专业性也越来越强，对测试人员的知识、能力要求也越来越高。
3、试阐述测试用例的设计原则有哪些。
（1）测试用例应覆盖三类事件。
①基本事件：参照软件规格说明书，按照需要实现的所有功能来编写，覆盖率 100%。
②备选事件：程序执行中的备选情况，按照功能点编写。
③异常事件：程序执行出错处理的路径，按照功能点编写。
在实际中，备选事件和异常事件的测试用例往往比基本事件的测试用例要多。
（2）用等价类划分方法设计基本的测试用例，将无限测试变成有限测试，这是减少工作量和提高测试效率最有效的方法。
（3）在任何情况下都应当有边界值测试用例，这种测试用例发现程序错误的能力最强。
（4）用错误推测法再追加一些测试用例，这需要依靠测试工程师的智慧和经验。
（5）应对照程序的逻辑，检查已设计测试用例的逻辑覆盖程度。如果没有达到要求的覆盖标准，应当再补充足够的测试用例。
（6）如果程序的功能说明中含有输入条件的组合情况，那么一开始就可以选用决策表驱动法和因果图法。

（7）对于参数配置类的软件来说，采用正交实验法设计较少的测试用例即可达到较好的测试效果。
（8）对于业务流清晰的系统来说，采用场景法来设计测试用例。

4、试分析测试用例为什么需要更新？

测试用例还需要不断更新和完善。主要原因有三个：

第一、在后续的测试过程中可能发现前面设计测试用例时考虑不周，需要补充完善；

第二、在软件交付使用后反馈了软件缺陷，而这些软件缺陷在测试时并没有发现，需要补充针对这些缺陷的测试用例；

第三、软件版本的更新及功能的新增等，要求测试用例也需要配套修改更新。