1.怎么做好文档测试?  
仔细阅读，跟随每个步骤，检查每个图形，尝试每个示例,检查文档的编写是否满足文档编写的目的,内容是否齐全，正确，完善.标记是否正确.

软件测试分哪2种方法？分别适合什么情况？

软件测试分2种：白盒测试和黑盒测试。白盒测试又称为结构测试、逻辑驱动测试或基于程序本身的测试，它着重于程序的内部结构及算法，通常不关心功能与性能指标；黑盒测试又称功能测试、数据驱动测试或基于规格说明的测试，它实际上是站在最终用户的立场，检验输入输出信息及系统性能指标是否符合规格说明书中有关功能需求及性能需求的规定  
2.白盒测试有几种方法?  
总体上分为静态方法和动态方法两大类。  
静态：关键功能是检查软件的表示和描述是否一致,没有冲突或者没有歧义  
动态：语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖。

3.系统测试计划是否需要同行审批，为什么?  
需要，系统测试计划属于项目阶段性关键文档，因此需要评审。

4.Alpha测试与beta的区别?  
Alpha测试在系统开发接近完成时对应用系统的测试；测试后仍然会有少量的设计变更。这种测试一般由最终用户或其它人员完成，不能由程序或测试员完成。  
Beta测试当开发和测试根本完成时所做的测试，最终的错误和问题需要在最终发行前找到。这种测试一般由最终用户或其它人员完成，不能由程序员或测试员完成。

5.比较负载测试，容量测试和强度测试的区别?  
负载测试：在一定的工作负荷下，系统的负荷及响应时间。  
强度测试：在一定的负荷条件下，在较长时间跨度内的系统连续运行给系统性能所造成的影响。  
容量测试：容量测试目的是通过测试预先分 析出反映软件 系统应用特征的某项指标的极限值（如最大并发用户数、数据库记录数等），系统在其极限值状  态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试 还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。容量测试的目的是使系统承受超额的数据容量来发现它是否能够正确处理。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。

6.测试结束的标准是什么？  
用例全部测试。  
覆盖率达到标准。  
缺陷率达到标准。  
其他指标达到质量标准

7.描述软件测试活动的生命周期？

测试周期分为计划、设计、实现、执行、总结。其中：  
计划：对整个测试周期中所有活动进行规划，估计工作量、风险，安排人力物力资源，安排进度等；  
设计：完成测试方案，从技术层面上对测试进行规划；  
实现：进行测试用例和测试规程设计；  
执行：根据前期完成的计划、方案、用例、规程等文档，执行测试用例。  
总结：记录测试结果，进行测试分析，完成测试报告。  
8.软件的缺陷等级应如何划分？  
A类—严重错误，包括以下各种错误： 1． 由于程序所引起的死机,非法退出 2． 死循环 3． 数据库发生死锁 4． 因错误操作导致的程序中断 5． 功能错误 6． 与数据库连接错误 7． 数据通讯错误  
B类—较严重错误，包括以下各种错误： 1． 程序错误 2． 程序接口错误 3． 数据库的表、业务规则、缺省值未加完整性等约束条件  
C类—一般性错误，包括以下各种错误： 1． 操作界面错误（包括数据窗口内列名定义、含义是否一致） 2． 打印内容、格式错误 3． 简单的输入限制未放在前台进行控制 4． 删除操作未给出提示 5． 数据库表中有过多的空字段  
D类—较小错误，包括以下各种错误： 1． 界面不规范 2． 辅助说明描述不清楚 3． 输入输出不规范 4． 长操作未给用户提示 5． 提示窗口文字未采用行业术语 6． 可输入区域和只读区域没有明显的区分标志

9. 当开发人员说不是BUG时,你如何应付？  
开发人员说不是bug，有2种情况，一是需求没有确定，所以我可以这么做，这个时候可以找来产品经理进行确认，需不需要改动，3方商量确定好后再看要 不要改。二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是BUG的依据是什么？如果被用户发现或出了问题，会有什么不良结果？ 程序员可能会给你很多理由，你可以对他的解释进行反驳。如果还是不行，那我可以给这个问题提出来,跟开发经理和测试经理进行确认,如果要修改就改,如果不 要修改就不改。其实有些真的不是bug，我也只是建议的方式写进TD中，如果开发人员不修改也没有大问题。如果确定是bug的话，一定要坚持自己的立场， 让问题得到最后的确认。

10．你为什么想离开目前的职务？  
因为公司运作情况并不理想，公司需要调整部门体系，公司考虑到缩减部门人员，所以大批量的裁员（有6，7个），这是我的第一份工作，对公司也有较深的 感情，因为在这里我找到了职业理想（就是测试），所以公司需要精简人员，我自愿退出。虽然很舍不得，但我将会有新的发挥能力的舞台。

11．您认为做好测试用例设计工作的关键是什么？  
白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果  
黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

12. 请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。  
黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。  
白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。  
软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序 的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：  
1、是否有不正确或遗漏的功能？  
2、在接口上，输入是否能正确的接受？能否输出正确的结果？  
3、是否有数据结构错误或外部信息（例如数据文件）访问错误？  
4、性能上是否能够满足要求？  
5、是否有初始化或终止性错误？  
软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计 或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测 试。白盒测试主要是想对程序模块进行如下检查：  
1、对程序模块的所有独立的执行路径至少测试一遍。  
2、对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测一遍。  
3、在循环的边界和运行的界限内执行循环体。  
4、测试内部数据结构的有效性，等等。  
单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件（或者场景）下某个特定函数的行为。  
单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。  
集成测试（也叫组装测试，联合测试）是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这 一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进 程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起测试。  
系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。（常见的联调测试）  
系统测试的目的是对最终软件系统进行全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。  
验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。  
验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。