（1）不属于计算机控制器中的部件。

（1） A.指令寄存器 IR B.程序计数器 PC

C.算术逻辑单元 ALU D.程序状态字寄存器 PSW

**【答案】C**

**【解析】**

控制器是分析和执行指令的部件，也是统一指挥和控制计算机各个部件按时序协调操作的部件。控制器的组成包含如下部分：程序计数器PC、指令寄存器IR、指令译码器、时序部件、微操作控制信号形成部件PSW、中断机构。故C答案的算术逻辑单元ALU不属于控制器，是运算器。

在 CPU 与主存之间设置高速缓冲存储器 Cache 的目的是为了（2）。

（2） A.扩大主存的存储容量 B.提高 CPU 对主存的访问效率

C.既扩大主存容量又提高存取速度 D.提高外存储器的速度

**【答案】B**

**【解析】**

为了提高CPU对主存的存取速度，又不至于增加很大的成本。现在，通常在CPU与主存之间设置高速缓冲存储器（Cache），其目的就在于提高速度而不增加很大代价。同时，设置高速缓冲存储器并不能增加主存的容量。

下面的描述中，（3）不是 RISC 设计应遵循的设计原则。

（3） A.指令条数应少一些

B.寻址方式尽可能少

C.采用变长指令，功能复杂的指令长度长而简单指令长度短

D.设计尽可能多的通用寄存器

**【答案】C**

**【解析】**

在设计RISC时，需要遵循如下一些基本的原则。

（1）指令条数少，一般为几十条指令。

（2）寻址方式尽可能少。

（3）采用等长指令，不管功能复杂的指令还是简单的指令，均用同一长度。

（4）设计尽可能多的通用寄存器。

因此，采用变长指令，功能复杂的指令长度长而简单指令长度短不是应采用的设计原则。

系统响应时间和作业吞吐量是衡量计算机系统性能的重要指标。对于一个持续处理业务的系统而言，其（4）。

（4） A.响应时间越短，作业吞吐量越小 B.响应时间越短，作业吞吐量越大

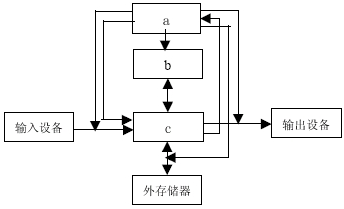
C.响应时间越长，作业吞吐量越大 D.响应时间不会影响作业吞吐量

**【答案】B**

**【解析】**

系统响应时间是指用户发出完整请求到系统完成任务给出响应的时间间隔。作业吞吐量是指单位时间内系统完成的任务量。若一个给定系统持续地收到用户提交的任务请求，则系统的响应时间将对作业吞吐量造成一定影响。若每个任务的响应时间越短，则系统的空闲资源越多，整个系统在单位时间内完成的任务量将越大；反之，若响应时间越长，则系统的空闲资源越少，整个系统在单位时间内完成的任务量将越小。

计算机各功能部件之间的合作关系如下图所示。假设图中虚线表示控制流，实线表示数据流，那么a、b和c分别表示（5）。



（5） A.控制器、内存储器和运算器 B.控制器、运算器和内存储器

C.内存储器、运算器和控制器 D.内存储器、控制器和运算器

**【答案】B**

**【解析】**

在一台计算机中，有以下6种主要的部件。

控制器（Control unit）：统一指挥并控制计算机各部件协调工作的中心部件，所依据的是机器指令。

运算器（亦称为算术逻辑单元，Arithmetic and Logic Unit，ALU）：对数据进行算术运算和逻辑运算。

内存储器（Memory或Primary storage，简称内存）：存储现场操作的信息与中间结果，包括机器指令和数据。

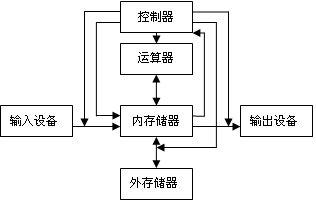
外存储器（Secondary storage或Permanent storage，简称外存）：存储需要长期保存的各种信息。

输入设备（Input devices）：接收外界向计算机输出的信息。

输出设备（Output devices）：将计算即中的信息向外界输送。

现在的控制器和运算器是被制造在同一块超大规模集成电路中的，称为中央处理器，即CPU（Central Processing Unit）。CPU和内存，统称为计算机的系统单元（Peripherals，简称外设）。

计算机各功能部件之间的合作关系如图所示。



下面关于编程语言的各种说法中，（6）是正确的。

（6） A.由于 C 语言程序是由函数构成的，所以也属于函数型语言

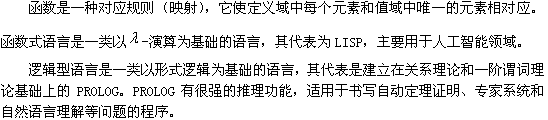
B.Smalltalk、C++、Java、C#都是面向对象语言

C.函数型语言适用于编写高速计算的程序，常用于超级计算机的模拟计算

D.逻辑型语言是在 Client/Server 系统中用于实现负载分散的程序语言

**【答案】B**

**【解析】**



在面向对象的语言中，（7）。

（7） A.类的实例化是指对类的实例分配存储空间

B.每个类都必须创建一个实例

C.每个类只能创建一个实例

D.类的实例化是指对类进行初始化

**【答案】A**

**【解析】**

类是用户定义的类型。与语言定义的基本类型一样，有了类型后，就可以定义（创建）该类型的变量，其含义是系统为变量分配存储空间。对于程序中定义的类，并不要求一定要创建其实例，对实例的数目也没有限制。创建类的实例时，系统需要为该实例分配存储空间。

在统一建模语言（UML）中，描述系统与外部系统及用户之间交互的图是（8）。

（8）A.类图 B.用例图 C.对象图 D.协作图

**【答案】B**

**【解析】**

·类图（class diagram）：展现了一组对象、接口、协作和它们之间的关系。在面向对象系统的建模中所建立的最常见的图就是类图。类图给出系统的静态设计视图。包含主动类的类图给出了系统的静态进程视图。

·对象图（object diagram）：展现了一组对象以及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图相同，这些图给出系统的静态设计视图或静态进程视图，但它们是从真实的或原型案例的角度建立的。

·用例图（use case diagram）：展现了一组用例、参与者（actor）以及它们之间的关系。用例图给出系统的静态用例视图。这些图对系统的行为进行组织和建模是非常重要的。

·序列图（sequence diagram）：是场景（scenano）的图形化表示，描述了以时间顺序组织的对象之间的交互活动。

·协作图（collaboration diagram或communication diagram）：强调收发消息的对象的结构组织。

序列图和协作图都是交互图（interaction diagram）。交互图展现了一种交互，它由一组对象和它们之间的关系组成，包括它们之间可能发送的消息。交互图关注系统的动态视图。序列图和协作图是同构的，它们之间可以相互转换。

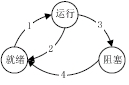
·状态图（state diagram）：展现了一个状态机，它由状态、转换、事件和活度组成。状态图关注系统的动态视图，它对于接口、类和协作的行为建模尤为重要，它强调对象行为的事件顺序。

·活动图（activity diagram）：是一种特殊的状态图，它展现了在系统内从一个活动到另一个活动的流程。活动图专注于系统的动态视图。它对于系统的功能建模特别重要，并强调对象间的控制流程。

·构件图（component diagram）：展现了一组构件之间的组织和依赖。构件图专注干系统的静态实现视图。它与类图相关，通常把构件映射为一个或多个类、接口或协作。

·部署图（deployment diagram）：展现了运行处理节点以及其中的构件的配置。部署图给出了体系结构的静态实施视图。它与构件图相关，通常一个节点包含一个或多个构件。

某系统的进程状态转换如下图所示。图中 1、2、3 和 4 分别表示引起状态转换时的不同原因。原因 4 是由于（9）；一个进程状态转换会引起另一个进程状态转换的是（10）。



（9） A.就绪进程被调度 B.运行进程执行了 P 操作

C.阻塞进程等待的事件发生了 D.运行进程时间片到了

（10）A.1→2 B.2→1 C.3→2 D.4→1

**【答案】C B**

**【解析】**

图中原因1是由于调度程序的调度引起；原因2是由于时间片用完引起；原因3是由于I/O请求引起，例如进程执行了P操作，由于申请的资源得不到满足进入阻塞队列；原因4是由于I/O完成引起的，例如某进程执行了V操作将信号量值减1，若信号量的值小于0，意味着有等待该资源的进程，将该进程从阻塞队列中唤醒使其进入就绪队列；因此试题（9）的正确答案是C。

试题（10）选项A“1→2”不可能，因为调度程序从就绪队列中调度一个进程投入运行，不会引起另外一个进程时间片用完；选项B“2→1”可能，因为当现运行进程的时间片用完，会引起调度程序调度另外一个进程投入运行；选项C“3→2”不可能，因为现运行进程由于等待某事件被阻塞，使得CPU空闲，此时调度程序会从处于就绪状态的进程中挑选一个新进程投入运行；选项D“4→1”不可能，一般一个进程从阻塞状态变化到就绪状态时，不会引起另一个进程从就绪状态变化到运行状态。

在操作系统中，虚拟设备通常采用（11）设备来模拟低速设备（如打印机）。

（11）A.Spooling 技术，利用磁带 B.Spooling 技术，利用磁盘

C.脱机批处理技术，利用磁盘 D.通道技术，利用磁带

**【答案】B**

**【解析】**

所谓Spooling技术实际上是用一类物理设备模拟另一类物理设备的技术，是使独占使用的设备变成多台虚拟设备的一种技术，也是一种速度匹配技术。Spooling系统是由“预输入程序/输出程序”和“井管理程序”以及输入和输出井组成的。其中，输入井和输出井是为了存放从输入设备输入的信息以及作业执行的结果，系统在磁盘上开辟的存储区域。

某文件管理系统在磁盘上建立了位示图(bitmap)，记录磁盘的使用情况。若系统中字长为 32 位，磁盘上的物理块依次编号为：0、1、2、…，那么 8192 号物理块的使用情况在位示图中的第（12）个字中有所描述。

（12）A.256 B.257 C.512 D.1024

**【答案】B**

**【解析】**

文件管理系统是在外存上建立一张位示图，记录文件存储器的使用情况。每一位对应文件存储器上的一个物理块，取值0和1分别表示空闲和占用，由于系统中字长为32位，所以每个字可以表示32个物理块的使用情况。又因为文件存储器上的物理块依次编号为：0，1，2，…．因此8192号物理块在位示图中的第257个字中描述。

在数据库管理系统中，（13）不属于安全性控制机制。

（13）A.完整性约束 B.视图 C.密码验证 D.用户授权

**【答案】A**

**【解析】**

数据库系统的安全措施主要有权限机制、视图机制以及数据加密三个方面。通过权限机制，限定用户对数据的操作权限，把数据的操作限定在具有指定权限的用户范围内在标准SQL中定义了授权语句GRANT来实现权限管理。通过建立用户视图，用户或应用程序只能通过视图来操作数据，保证了视图之外的数据的安全性。对数据库中的数据进行加密，可以防止数据在存储和传输过程中失密。

数据库的完整性是指数据的正确性和相容性。如学生的性别只能是男或女，百分制的成绩只能取0～100之间的整数值等。为防止错误数据进入数据库，DBMS提供了完整性约束机制，通过对数据库表结构进行约束，当对数据进行修改时由系统对修改数据进行完整性检查，将错误数据拒绝于数据库之外。

因此在数据库管理系统中，完整性约束不属于安全性控制机制，而属于完整性约束机制。

某供应商数据库中的供应关系为 SPJ（供应商号，零件号，工程号，数量），如下命令查询某工程至少用了3家供应商（包含 3 家）供应的零件的平均数量，并按工程号的降序排列。

SELECT 工程号，（14）FROM SPJ

GROUP BY 工程号

（15）

ORDER BY 工程号 DESC；

（14）A.AVG（数量）At 平均数量 B.AVG（数量）AS 平均数量

C.平均数量 At AVG（数量） D.平均数量 AS AVG（数量）

（15）A.HAVING COUNT（DISTINCT（供应商号））>2

B.Where COUNT（供应商号）>2

C.HAVING（DISTINCT（供应商号））>2

D.Where 供应商号 >2

**【答案】B A**

**【解析】**

试题（14）正确的答案是选项B，因为SQL提供可为关系和属性重新命名的机制，这是通过使用具有如下形式的as子句来实现的。

Old-name as new-name

as子句即可出现在select子旬，也可出现在from子句中。显然选项B满足上述规则：

试题（15）正确的答案是选项A，因为根据题意“某工程至少用了3家供应商（包含3家）供应的零件”，应该按照工程号分组，而且应该加上条件供应商的数目。但是需要注意的是，一个工程项目可能用了同一个供应商的多种零件，因此，在统计供应商数的时候需要加上DISTINCT，以避免重复统计导致错误的结果。

假如按工程号JNO＝’J1’分组，结果如下表所示。



从上表可以看出，如果不加DISTINCT，统计的数为7；而加了DISTINCT，统计的数是5。

设关系模式 R（A， B， C），传递依赖指的是（16）；下列结论错误的是（17）。

（16）A.若 A→B，B→C，则 A→C B.若 A→B，A→C，则 A→BC

C.若 A→C，则 AB→C D.若 A→BC，则 A→B，A→C

（17）A.若 A→BC，则 A→B，A→C B.若 A→B，A→C，则 A→BC

C.若 A→C，则 AB→C D.若 AB→C，则 A→C，B→C

**【答案】A D**

**【解析】**

试题（16）正确的答案是选项A，因为选项A满足传递规则；试题（17）正确的答案是选项D，因为选项A满足分解规则；选项B是合并规则：选项C中，A→C成立，则给其决定因素A再加上其他冗余属性B也成立；选项D不成立，反例：如AB为学号和课程号，C为成绩，则学号、课程号→成绩成立，但学号→成绩不成立。同样，也可以用证明的方法来判定。

软件开发人员可以用（18）软件编写和修改程序。

（18）A.预处理 B.文本编辑 C.链接 D.编译

**【答案】B**

**【解析】**

软件开发人员可以用文本编辑软件编写和修改程序。

统一过程（UP）是一种用例驱动的迭代式增量开发过程，每次迭代过程中主要的工作流包括捕获需求、分析、设计、实现和测试等。这种软件过程的用例图(Use Case Diagram)是通过（19）得到的。

（19）A.捕获需求 B.分析 C.设计 D.实现

**【答案】A**

**【解析】**

统一过程的基本特征是用例驱动、以架构为中心的迭代式增量开发。其核心的工作流包括捕获用户需求、分析、设计、实现和测试等。首先，开发人员通过和用户的沟通、了解，捕获用户需求并制作软件用例图（用况图），从而得到软件的用况模型。然后分析并设计满足这些用例的系统，得到分析模型、设计模型和实施模型，进而实现该系统，最后通过测试模型来验证系统是否满足用况中描述的功能。需求捕获过程中得到的用例将各个核心工作流结合为一个整体，驱动整个软件开发过程。

关于原型化开发方法的叙述中，不正确的是（20）。

（20）A.原型化方法适应于需求不明确的软件开发

B.在开发过程中，可以废弃不用早期构造的软件原型

C.原型化方法可以直接开发出最终产品

D.原型化方法利于确认各项系统服务的可用性

**【答案】C**

**【解析】**

原型化软件开发方法的基本思想是软件开发人员对用户提出的需求和问题进行总结，就系统的主要需求取得一致意见后，构造一个软件原型（原型是软件的一个早期版本，通常反映最终软件的部分重要特性，原型应该是可以运行和修改的），使用户在试用原型过程中得到感受和启发，并做出反应和评价。然后开发者根据用户的意见对原型进行改进，使之逐步完善，直到用户对系统完全满意为止。这种开发方法的优点是需求表示清楚，用户满意度较高、可降低开始风险和开发成本。所以原型化方法特别适应于原始需求不明确的软件，因为通过用户的不断使用和体验并提出的评价，使得不断修改的原型逐步达到用户要求。通常，软件开发过程中会得到多个软件原型，只有得到用户认可的才是最终的产品。

基本思想：开发人员对用户提出的问题进行总结，就系统的主要需求取得一致意见后，开发一个原型（原型是由开发人员与用户合作，共同确定系统的基本要求和主要功能，并在较短时间内开发的一个实验性的、简单易用的小型系统。原型应该是可以运行的，可以修改的。）并运行之，然后反复对原型进行修改，使之逐步完善，直到用户对系统完全满意为止。

优点：

（1）需求表示清楚，用户满意度较高。

（2）降低开始风险和开发成本。

缺点：

（1）原型法不适用于开发大型的信息系统。

（2）系统难于维护。

（3）如果用户合作不好，盲目纠错，会拖延开发进程。

适用范围：

（1）用户需求不清，管理及业务不稳定，需求经常变化。

（2）规模小，不太复杂。

（3）开发信息系统的最终用户界面。

CMM 模型将软件过程的成熟度分为 5 个等级。在（21）使用定量分析来不断地改进和管理软件过程。

（21）A.优化级 B.管理级 C.定义级 D.可重复级

**【答案】A**

**【解析】**

CMM的分级结构及其过程描述

（1）初始级：软件过程的特点是无秩序或说无定规的，有时甚至是混乱的。软件过程定义几乎处于无章法、无步骤可循的状态，软件产品所取得的成功往往依赖于极个别人的努力和机遇。

（2）可重复级：已建立了基本的项目管理过程，可用于对成本、进度和功能特性进行跟踪。对类似的应用项目，有章可循并能重复以往所取得的成功。

（2）可重复级：已建立了基本的项目管理过程，可用于对成本、进度和功能特性进行跟踪。对类似的应用项目，有章可循并能重复以往所取得的成功。

（4）已管理级：软件过程和产品质量有详细的度量标准。软件过程和产品质量得到了定量的认识和控制。

（5）优化级：通过对来自过程、新概念和新技术等方面的各种有用信息的定量分析，能够不断地、持续地对促进过程进行改进。

除第一级外，每一级都设定了一组目标，如果达到了这组目标，则表明达到了这个成熟级别，自然可以向上一更为成熟的高一级别迈进。CMM体系不主张跨级别的进化，因为从第二级开始，每一个低级别的实现均是更高级别实现的基础。

软件（22）的提高，有利于软件可靠性的提高。

（22）A.存储效率 B.执行效率 C.容错性 D.可移植性

**【答案】C**

**【解析】**

软件可靠性是指软件按照用户的要求和设计的目标，执行其功能的正确程度，所以一个可靠的软件应当是正确的、完整的、一致的和健壮的。通常，软件可靠性受到其他质量特性的影响，在4个备选项中，A、B、D都是不利于软件可靠性的因素。

序言性注释是指在每个程序或模块开头的一段说明，起辅助理解程序的作用，一般包括：程序的表示、名称和版本号；程序功能描述；接口与界面描述；输入/输出 数据说明；开发历史；与运行环境有关的信息等。下列叙述中不属于序言性注释的是（23）。

（23）A.程序对硬件、软件资源的要求

B.重要变量和参数说明

C.嵌入在程序中的 SQL 语句

D.程序开发的原作者、审查者、修改者、编程日期等

**【答案】C**

**【解析】**

序言性注释给出程序的整体说明，对于理解程序具有引导作用。“嵌入在程序中的SQL语旬”是数据库系统提供的一种应用程序接口，不是注释。

LOC 是软件规模的一种量度，它表示（24）。

（24）A.软件功能数 B.源代码行数 C.每单位成本数 D.工作量

**【答案】B**

**【解析】**

LOC是Lines of Codes的缩写，指构成软件的源代码行数。LOC主要用于估算软件成本和规模。软件估算的另外一个传统参数是功能数目。

面向对象分析与设计是面向对象软件开发过程中的两个重要阶段，下列活动中，（25）不属于面向对象分析阶段。

（25）A.构建分析模型 B.识别分析类

C.确定接口规格 D.评估分析模型

**【答案】C**

**【解析】**

面向对象的软件开发过程包括分析、系统设计、开发类、组装测试和应用维护等。其中分析过程包括问题域分析、应用分析，此阶段主要识别对象及对象之间的关系，最终形成软件的分析模型，并进行评估。设计阶段主要构造软件总的模型，实现相应源代码，在此阶段，需要发现对象的过程，确定接口规格。

面向对象分析需要找出软件需求中客观存在的所有实体对象（概念），然后归纳、抽象出实体类。（26）是寻找实体对象的有效方法之一。

（26）A.会议调查 B.问卷调查 C.电话调查 D.名词分析

**【答案】D**

**【解析】**

面向对象分析中寻找实体类中，名词分析法是通过分析需求文档中出现的名词来寻找实体类，它是面向对象分析的有效方法之一。而会议调查、问卷调查和电话调查的目的并非寻找实体类，而是得到软件系统的需求。

在“模型－视图－控制器”（MVC）模式中，（27）主要表现用户界面，（28）用来描述核心业务逻辑。

（27）A.视图 B.模型 C.控制器 D.视图和控制器

（28）A.视图 B.模型 C.控制器 D.视图和控制器

**【答案】A B**

**【解析】**

MVC模式是一个复杂的架构模式，其实现也显得非常复杂。

视图(View)代表用户交互界面，对于Web应用来说，可以概括为HTML界面，但有可能为XHTML、XML和Applet。

模型(Model)：就是业务流程/状态的处理以及业务规则的制定。业务模型的设计可以说是MVC最主要的核心。

控制(Controller)可以理解为从用户接收请求， 将模型与视图匹配在一起，共同完成用户的请求。

在进行面向对象设计时，采用设计模式能够（29）。

（29）A.复用相似问题的相同解决方案 B.改善代码的平台可移植性

C.改善代码的可理解性 D.增强软件的易安装性

**【答案】A**

**【解析】**

因为模式是一种指导，在一个良好的指导下，有助于你完成任务，有助于你作出一个优良的设计方案，达到事半功倍的效果，而且会得到解决问题的最佳办法。采用设计模式能够复用相似问题的相同解决方案，加快设计的速度，提高了一致性。

下面给出了四种设计模式的作用：

外观（Façade）：为子系统中的一组功能调用提供一个一致的接口，这个接口使得这一子系统更加容易使用；

装饰(Decorate)：当不能采用生成子类的方法进行扩充时，动态地给一个对象添加一些额外的功能；

单件（Singleton）：保证一个类仅有一个实例，并提供一个访问它的全局访问点；

模板方法（Template Method）：在方法中定义算法的框架，而将算法中的一些操作步骤延迟到子类中实现。

请根据下面叙述的场景选用适当的设计模式。若某面向对象系统中的某些类有且只 有一个实例，那么采用（30）设计模式能够有效达到该目的；该系统中的某子模块需 要为其他模块提供访问不同数据库系统（Oracle、SQL Server、DB2 UDB 等）的功能，这些数据库系统提供的访问接口有一定的差异，但访问过程却都是相同的，例如，先连 接数据库、再打开数据库、最后对数据进行查询，（31）设计模式可抽象出相同的数据 库访问过程；系统中的文本显示类（TextView）和图片显示类（PictureView）都继承 了组件类（Component），分别显示文本和图片内容，现需要构造带有滚动条、或者带有黑色边框、或者既有滚动条又有黑色边框的文本显示控件和图片显示控件，但希望最多 只增加三个类，（32）设计模式可以实现该目的。

（30）A.外观 B.装饰 C.单件 D.模板方法

（31）A.外观 B.装饰 C.单件 D.模板方法

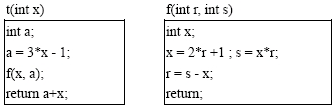
（32）A.外观 B.装饰 C.单件 D.模板方法

**【答案】C D B**

**【解析】**

面向对象系统中的某些类有且只有一个实例，该场景的描述与单件模式的定义相同。系统中的某子模块需要为其他模块提供访问不同数据库系统（Oracle、SQL Server、DB2 UDB等）的功能，这些数据库系统提供的访问接口有一定的差异，但访问过程却都是相同的。例如，先连接数据库，再打开数据库，最后对数据进行查询，该场景描述了对数据库进行操作的步骤是相同的。但是，具体的每个步骤根据不同的数据库系统会存在一定差异，例如数据库提供的接口函数不同，模板方法正是将步骤过程抽象出来，而每个具体操作步骤的差异留到具体的子类去实现。系统中的文本显示类（ Text View）和图片显示类（Picture View）都继承了组件类（Component），分别显示文本和图片内容，现需要构造带有滚动条、或者带有黑色边框、或者既有滚动条又有黑色边框的文本显示控件和图片显示控件，但希望最多只增加三个类，该场景限定了能够增加的类的数量。可以通过新增加三个类，分别继承组件类并实现给组件增加黑色边框、滚动条以及增加黑色边框和滚动条功能，因为文本显示类和图片显示类都属于组件类，因此，新增加的三个类能够给文本显示对象和图片显示对象增加额外的显示功能，该实现手段采用的就是装饰模式。

函数 t()、f()的定义如下所示。若调用函数 t()时传递给 x 的值为 3，并且调用函数 f()时，第一个参数采用传值（call by value）方式，第二个参数采用传引用（call by reference）方式，则函数 t()的返回值为（33）。



（33）A.35 B.24 C.22 D.11

**【答案】B**

**【解析】**

采用传值方式时，是将实际参数的值传递给形式参数，对形式参数值的修改不会影响实际参数。

采用引用方式时，是将实际参数的地址传递给形式参数，对形式参数进行修改，等同于是对实际参数进行修改。

程序设计语言中（34）。

（34）A.while循环语句的执行效率比do-while循环语句的执行效率高

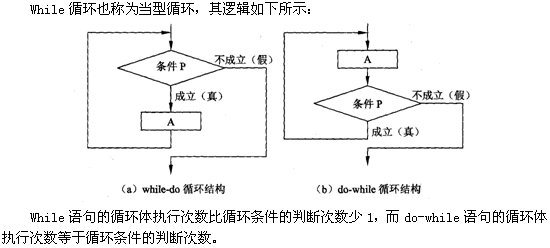
B.while循环语句的循环体执行次数比循环条件的判断次数多1，而do-while语句的循环体执行次数比循环条件的判断次数少1

C.while语句的循环体执行次数比循环条件的判断次数少1，而do-while语句的循环体执行次数比循环条件的判断次数多1

D.while语句的循环体执行次数比循环条件的判断次数少1，而do-while语句的循环体执行次数等于循环条件的判断次数

**【答案】D**

**【解析】**



两名以上的申请人分别就同样的发明创造申请专利的，专利权授权（35）。

（35）A.最先发明的人 B.最先申请的人 C.所有申请的人 D.协商后的申请人

**【答案】B**

**【解析】**

根据我国专利法第九条规定“两个以上的申请人分别就同样的发明创造申请专利的，专利权授予最先申请的人。”，针对两名以上的申请人分别就同样的发明创造申请专利，专利权应授予最先申请的人。

下列叙述中（36）是正确的。

（36）A.压迫测试：提供条件任软件发挥，最大限度地发掘软件的能力

B.重负测试：使软件在不够理想的条件下运行，观察软件对外部资源的要求和依赖的程度

C.重复测试：不断执行同样的操作，这种反复测试的主要原因是看内存是否不足

D.完整 C/S 体系结构测试，只包括网络运行和性能测试

**【答案】C**

**【解析】**

压迫测试：使软件在不够理想的条件下运行，观察软件对外部资源的要求和依赖的程度。

重负测试：提供条件任软件发挥，最大限度地发掘软件的能力。

重复测试：又称为疲劳强度测试，不断执行同样的操作，这种反复测试的主要原因是看内存是否不足或者是否有内存泄漏。

完整 C/S 体系结构测试：包括网络运行测试、服务器端性能测试和客户端性能测试。

根据 ANSI／IEEE829 标准，以下（37）属于《测试程序说明》中程序步骤的内容。

① 启动 ② 目的 ③ 日志 ④ 设置

（37）A.①、②、③ B.①、③、④ C.②、③ D.①、②、③、④

**【答案】B**

**【解析】**

根据ANSI/IEEE829标准，《测试程序说明》中程序步骤包括的内容主要有：

（1）日志：测试人员按测试要求记录程序执行过程，所有必须填写的项都必须填写，包括问题的记录。

（2）设置：测试人员必须安装Windows 98的干净副本，使用测试用具Tool-A和Tool-B等。

（3）启动：启动Windows 98，单击“开始”按钮，选择程序→附件→计算器。

（4）程序：用键盘输入每个测试用例并比较结果。

以下关于功能测试用例的意义的叙述，正确的是（38）。

① 避免盲目测试并提高测试效率

② 令软件测试的实施重点突出、目的明确

③ 在回归测试中无需修正测试用例便可继续开展测试工作

④ 测试用例的通用化和复用化使软件测试易于开展

（38）A.①、②、③ B.①、③ C.②、③ D.①、②、④

**【答案】D**

**【解析】**

功能测试用例的意义可以概括如下：

（1）为达到最佳的测试效果或高效地揭露隐藏的错误而精心设计的少量测试数据。

（2）测试用例的使用令软件测试的实施重点突出、目的明确。

（3）一个好的测试用例在于它能发现至今未发现的错误。

但是，在回归测试中，为了满足测试需求，必须修正测试用例。

用等价类法划分 Windows 文件名称，应该分成（39）个等价区间。

（39）A.2 B.3 C.4 D.6

**【答案】C**

**【解析】**

用用等价类法划分 Windows 文件名称，应该分成4个等价区间，分别是过长名称、过短名称、合法长度合法字符、合法长度非法字符。

用边界值分析法，假定 X 为整数，10≤X≤100，那么 X 在测试中应该取（40）边界值。

（40）A.X=10，X=100 B.X=9，X=10，X=100，X=101

C.X=10，X=11，X=99，X=100 D.X=9，X=10，X=50，X=100

**【答案】B**

**【解析】**

用边界值分析法，如以A和B为边界，测试用例应该包括A和B，以及略大于A和略小于B的值。在本题中，因为 X 为整数，且10≤X≤100，那么 X 在测试中应该取：X=9，X=10，X=100，X=101。

（41）不是易用性测试包括的内容。

（41）A.安装测试 B.界面测试 C.菜单测试 D.文档测试

**【答案】D**

**【解析】**

易测试性包括安装测试、功能易用性测试、界面测试和辅助系统测试

（42）不是文档测试包括的内容。

（42）A.合同文档 B.开发文档 C.管理文档 D.用户文档

**【答案】A**

**【解析】**

根据文档产生、使用的范围的不同，可以将其分为三大类：

（1）开发文档：为开发工作提供支持的各种文档，其读者群主要针对开发人员。其中主要包括需求规格说明书、数据要求规格说明书、概要设计说明书、详细设计说明书、项目开发计划等；

（2）管理文档：为项目的开发管理提供支持的各种文档，其读者群主要针对管理人员，其中主要包括可行性研究报告、项目开发计划、测试计划、技术报告、开发进度记录、项目开发总结报告等；

（3）用户文档：向用户传达各种与开发相关、与产品相关的信息，其读者群主要针对最终用户。其中主要包括用户手册、操作手册、维护修改建议书、软件需求说明书等。

针对用户手册的测试，（43）描述不正确。

（43）A.准确地按照手册的描述使用程序 B.检查每条陈述

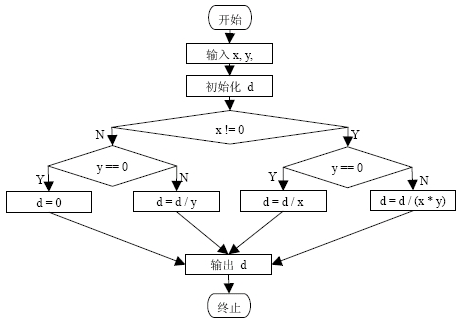
C.修改错误设计 D.查找容易误导用户的内容

**【答案】C**

**【解析】**

用户手册测试的内容包括准确地按照手册的描述使用程序、尝试每一条建议、检查每条陈述、查找容易误导用户的内容。

阅读下列流程图：当用判定覆盖法进行测试时，至少需要设计（44）个测试用例。



（44）A.2 B.4 C.6 D.8

**【答案】B**

**【解析】本题考查判定覆盖法。**

判定覆盖法是程序中每个判定的结果至少都获得一次“真”值和“假”值。在本题中，N与Y可以分别取“真”值和“假”值，所以需要设计4个测试用例。

Web应用链接测试不包括（45）。

（45）A.无链接指向的页面 B.错误的链接

C.客户端与服务器端的链接速率 D.不存在的页面文件

**【答案】C**

**【解析】**

Web应用链接测试包括无链接指向的页面、错误的链接、不存在的页面文件。客户端与服务器端的链接速率由Web性能测试获取。

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（46）。

（46）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

**【答案】B**

**【解析】**

学生年龄的输入范围为16～40，根据等价类划分技术，划分正确的是1 个有效等价类，2 个无效等价类，分别如下。

1 个有效等价类：16～40之间。

2 个无效等价类：小于16，大于40。

以下各项中，（47）属于安装测试应关注的内容。

①安装手册的评估 ②安装选项和设置的测试 ③安装顺序测试 ④修复安装测试与卸载测试

（47）A.①、②、③ B.③、④ C.②、③、④ D.①、②、③、④

**【答案】D**

**【解析】**

安装测试的内容包括安装手册的评估、安装的自动化程度测试、安装选项和设置的测试、安装过程的中断测试、安装顺序测试、多环境安装测试、安装的正确性测试、修复安装测试与卸载测试。

下面关于软件测试的说法，（48）是错误的。

（48）A.软件测试就是程序测试

B.软件测试贯穿于软件定义和开发的整个期间

C.需求规格说明、设计规格说明都是软件测试的对象

D.程序是软件测试的对象

**【答案】A**

**【解析】**

需求规格说明、概要设计规格说明、详细设计规格说明、程序都是软件测试的对象。软件测试贯穿于软件定义和开发的整个周期。

关于白盒测试与黑盒测试的最主要区别，正确的是（49）。

（49）A.白盒测试侧重于程序结构，黑盒测试侧重于功能

B.白盒测试可以使用测试工具，黑盒测试不能使用工具

C.白盒测试需要程序员参与，黑盒测试不需要

D.黑盒测试比白盒测试应用更广泛

**【答案】A**

**【解析】**

动态测试指通过运行程序发现错误，分为黑盒测试法和白盒测试法。

黑盒测试把被测试对象看成一个黑盒子，测试人员完全不考虑程序的内部结构和处理过程，只在软件的接口处进行测试，依据需求规格说明书，检查程序是否满足功能要求。因此，黑盒测试又称为功能测试或数据驱动测试，使用这种方法，为了做到穷尽测试，至少必须对所有输入数据的各种可能值的排列组合都进行测试。即黑盒测试使用所有有效和无效的输入数据来测试程序是不现实的，所以黑盒测试同样不能做到穷尽测试。只能选取少量最有代表性的输入数据，以期用较少的代价暴露出较多的程序错误。

白盒测试又称为穷举路径测试或逻辑驱动测试，把测试对象看作一个打开的盒子，测试人员须了解程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试，检验内部控制结构和数据结构是否有错，实际的运行状态与预期的状态是否一致。由于白盒测试是结构测试，所以被测对象基本上是源程序，以程序的内部逻辑为基础设计测试用例。需要根据概要设计和详细设计说明文档生成用于白盒测试的测试用例。

软件测试按实施组织分，测试应该包括以下的（50）。

① 开发方测试 ② 用户方测试 ③ 第三方测试 ④ 验收测试 ⑤ 确认测试

（50）A.①、②、③ B.③、④、⑤ C.①、②、④ D.①、②、③、④、⑤

**【答案】A**

**【解析】**

软件测试按实施组织分，测试应该包括开发方测试、用户方测试、第三方测试。

以下各项中，（51）属于需求说明书的评测内容。

① 系统定义的目标是否与用户的要求一致

② 设计的约束条件或限制条件是否符合实际

③ 是否考虑过软件需求的其他方案

④ 软件的行为与它必须处理的信息、必须完成的功能是否一致

（51）A.①、②、④ B.①、③、④

C.②、③、④ D.①、②、③、④

**【答案】D**

**【解析】**

需求说明书评测作为需求分析阶段工作的复查手段，应该对功能的正确性、完整性和清晰性，以及其他需求给予评测。评测的主要内容如下：

（1）系统定义的目标是否与用户的要求一致。

（2）系统需求分析阶段提供的文档资料是否齐全。

（3）文档中的所有描述是否完整、清晰、准确地反映用户要求。

（4）与所有其他系统成分的重要接口是否都已经描述。

（5）被开发项目的数据流与数据结构是否足够、确定。

（6）所有图表是否清楚、在不补充说明时能否理解。

（7）主要功能是否已经包括在规定的软件范围之内，是否都已经充分说明。

（8）软件的行为和它必须处理的信息、必须完成的工程是否一致。

（9）设计的约束条件或限制条件是否符合实际。

（10）是否考虑了开发的技术风险。

（11）是否考虑过软件需求的其他方案。

（12）是否考虑过将来可能会提出的软件需求。

（13）是否详细制定了检验标准，它们能否对系统定义成功进行确认。

（14）有没有遗漏、重复或不一致的地方。

（15）用户是否审查了初步的用户手册或原型。

（16）项目开发计划中的估算是否受到了影响。

关于对第三方测试的描述，正确的观点是（52）。

（52）A.既不是用户，也不是开发人员所进行的测试就是第三方测试

B.第三方测试也称为独立测试，是由相对独立的组织进行的测试

C.第三方测试是在开发方与用户方的测试基础上进行的验证测试

D.第三方测试又被称为β测试

**【答案】B**

**【解析】**

第三方测试是介于软件开发方和用户方之间的测试组织的测试，第三方测试也称为独立测试。软件质量工程强调开展独立验证和确认（IV&V）活动。IV&V是由在技术、管理和财务上与开发组织具有规定程度独立的组织执行验证和确认过程。软件第三方测试也是由在技术、管理和财务上与开发方和用户方相对独立的组织进行的软件测试。一般情况下是模拟用户真实应用环境进行软件确认测试。

以下（53）属于在GB/T 18905《评价者用的过程》中定义软件评价过程的特性。

①可重复性 ②可再现性 ③公正性 ④客观性 ⑤合理性

（53）A.①、②、④ B.①、②、③、④

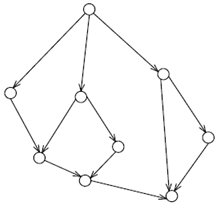
C.③、④、⑤ D.①、②、③、④、⑤

**【答案】B**

**【解析】**

在GB/T 18905《评价者用的过程》中定义软件评价过程的特性包括可重复性、可再现性、公正性、客观性。

以下控制流程图的环路复杂性 V(G)等于（54）。



（54）A.4 B.5 C.6 D.1

**【答案】B**

**【解析】**

控制流程图的环路复杂性 V(G)等于：

（1）控制流程图中的区域个数。

（2）边数-结点数+2。

（3）判定数+1。

通过疲劳强度测试，最容易发现（55）问题。

（55）A.并发用户数 B.内存泄漏 C.系统安全性 D.功能错误

**【答案】B**

**【解析】**

重复测试又称为疲劳强度测试，不断执行同样的操作，这种反复测试的主要原因是看内存是否不足或者是否有内存泄漏。

针对下列程序段，对于(A，B，C)的取值，以下（56）测试用例组合能够满足语句覆盖的要求。

IF ((A+10)=2 OR (B-20)<3) THEN C=0

IF ((A+30)>10 AND (C-30)<0) THEN B=30

（56）A.(2，30，1) B.(-20，0，30) C.(-30，20，30) D.(2，20，3)

**【答案】D**

**【解析】**

语句覆盖要求每一个可执行的语句至少执行1次。这段程序中有2个语句（“C=0”和“B=30”），如果A选择2，B选择20，C选择3，则2个语句都会被执行到。

针对下列程序段，对于(A，B)的取值，以下（57）测试用例组合能够满足条件覆盖的要求。

IF ((A-10)=20 AND (B+20)>10) THEN C=0

IF ((A-30)<10 AND (B-30)<0) THEN B=30

①A=50 B=-10 ②A=40 B=40 ③A=30 B=-10 ④A=30 B=30

（57）A.①② B.③④ C.①④ D.②④

**【答案】C**

**【解析】**



针对逻辑覆盖有下列叙述，（58）是不正确的。

（58）A.达到 100％DC 要求就一定能够满足 100％SC 的要求

B.达到 100％CC 要求就一定能够满足 100％SC 的要求

C.达到 100％CDC 要求就一定能够满足 100％SC 的要求

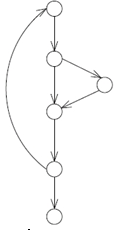
D.达到 100％MCDC 要求就一定能够满足 100％SC 的要求

**【答案】B**

**【解析】**

逻辑覆盖是以程序内部逻辑为基础的测试技术，常用的逻辑覆盖从低到高分别是：语句覆盖（SC）、判定覆盖（DC）、条件覆盖（CC）、判定/条件覆盖（CDC）、条件组合覆盖（MCC）、修正条件/判定覆盖（MCDC）。

以下所示程序控制流程图中有（59）条线性无关的基本路径。



（59）A.1 B.2 C.3 D.4

**【答案】C**

**【解析】**

线性无关的基本路径就是控制流程图中的圈复杂度（环路复杂性）， 控制流程图的环路复杂性 V(G)等于：

（1）控制流程图中的区域个数。

（2）边数-结点数+2。

（3）判定数+1。

因此本题的答案选C。

下列叙述中，（60）是正确的。

（60）A.白盒测试又称为逻辑驱动测试

B.穷举路径测试可以查出程序中因遗漏路径而产生的错误

C.一般而言，黑盒测试对结构的覆盖比白盒测试高

D.必须根据软件需求说明文档生成用于白盒测试的测试用例

**【答案】A**

**【解析】**

动态测试指通过运行程序发现错误，分为黑盒测试法和白盒测试法。

黑盒测试把被测试对象看成一个黑盒子，测试人员完全不考虑程序的内部结构和处理过程，只在软件的接口处进行测试，依据需求规格说明书，检查程序是否满足功能要求。因此，黑盒测试又称为功能测试或数据驱动测试，使用这种方法，为了做到穷尽测试，至少必须对所有输入数据的各种可能值的排列组合都进行测试。即黑盒测试使用所有有效和无效的输入数据来测试程序是不现实的，所以黑盒测试同样不能做到穷尽测试。只能选取少量最有代表性的输入数据，以期用较少的代价暴露出较多的程序错误。

白盒测试又称为穷举路径测试或逻辑驱动测试，把测试对象看作一个打开的盒子，测试人员须了解程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试，检验内部控制结构和数据结构是否有错，实际的运行状态与预期的状态是否一致。由于白盒测试是结构测试，所以被测对象基本上是源程序，以程序的内部逻辑为基础设计测试用例。需要根据概要设计和详细设计说明文档生成用于白盒测试的测试用例。

针对以下程序段，对于变量 c 的取值，至少需要（61）个测试用例才能够满足语句覆盖的要求。

c = ((u8\_t \*)q->payload)[i];

switch (c)

{

case SLIP\_END:

sio\_send(SLIP\_ESC， netif->state);

sio\_send(SLIP\_ESC\_END， netif->state);

break;

case SLIP\_ESC:

sio\_send(SLIP\_ESC， netif->state);

sio\_send(SLIP\_ESC\_ESC， netif->state);

break;

default:

sio\_send(c， netif->state);

break;

}

（61）A.4 B.3 C.2 D.1

**【答案】B**

**【解析】**

语句覆盖法要求每个可执行语句至少执行一次。本程序段中开关语句共有三个分支，对于变量c的取值，需要3个测试用例能够满足语句覆盖的要求。

针对以下 C 语言程序段，对于(MaxNum，Type)的取值，至少需要（62）个测试用例能够满足判定覆盖的要求。

while ( MaxNum-- > 0 )

{

if ( 10 == Type )

x = y \* 2;

else

if ( 100 == Type )

x = y + 10;

else

x = y - 20;

}

（62）A.5 B.4 C.3 D.2

**【答案】C**

**【解析】**

判定覆盖法要求程序中每个判定的结果至少都获得一次真值和一次假值。本题中共嵌套3个判定语句对于（MaxNum，Type）的取值，至少需要3个测试用例才能够满足判定覆盖的要求。

假设 A、B 为布尔变量，对于逻辑表达式（A&&B），至少需要（63）个测试用例才能完成 MCDC 覆盖。

（63）A.4 B.3 C.2 D.1

**【答案】B**

**【解析】**



广义的软件测试包括（64）。

（64）A.单元测试、集成测试、确认测试和系统测试

B.确认、验证和测试

C.需求评审、设计评审、单元测试和综合测试

D.开发方测试、用户测试和第三方测试

**【答案】B**

**【解析】**

广义的软件测试是由确认、验证和测试三个方面组成的。

确认是评估将要开发的软件产品是否正确无误、可行和有价值。确认意味着确保一个待开发软件是正确无误的，是对软件开发构想的检测。

验证是检测软件开发的每个阶段、每个步骤的结果是否正确无误，是否与软件开发各阶段的要求或期望的结果相一致。验证意味着确保软件会正确无误地实现软件的需求，开发过程是沿着正确的方向进行的。

GB/T 16260 将软件的内部（外部）质量属性划分为六大质量特性，分别是（65）。

（65）A.功能性，可靠性，易用性，效率，维护性和可移植性

B.功能性、可靠性、易用性、效率、稳定性和可移植性

C.功能性、可靠性、安全性、效率、易用性和可移植性

D.功能性、可靠性、兼容性、效率、稳定性和可移植性

**【答案】A**

**【解析】**

GB/T 16260将软件的内部（外部）质量属性划分为六大质量特性，分别是功能性、可靠性、易用性、效率、维护性和可移植性。

软件内部/外部质量模型中，以下（66）不是功能性包括的子特性。

（66）A.适合性 B.准确性 C.稳定性 D.互操作性

**【答案】C**

**【解析】**

功能性包括的子特性有适合性、准确性、互操作性、安全保密性、功能依从性。

《GB/T 18905 软件工程　产品评价》中确定的通用评价过程包括四个方面，其中有关“规定评价”部分包含的内容有（67）。

（67）A.选择度量、建立度量评定等级、确立评估准则

B.指定质量模型、选择度量、建立度量评定等级

C.选择度量、建立度量评定等级、制定评价计划

D.确定产品类型、选择度量、建立度量评定等级

**【答案】A**

**【解析】**

《GB/T 18905 软件工程 产品评价》中确定的通用评价过程包括四个方面，其中有关“规定评价”部分包含选择度量、建立度量评定等级、确立评估准则。

下列测试工具中，使用（68）执行自动化负载压力测试，使用（69）执行代码静态结构分析，使用（70）执行网络测试。

（68）A.SmartBits B.Logiscope C.Quick Test Professional D.LoadRunner

（69）A.SmartBits B.Logiscope C.Quick Test Professional D.LoadRunner

（70）A.SmartBits B.Logiscope C.Quick Test Professional D.LoadRunner

**【答案】D B A**

**【解析】**

负载压力测试主要是度量应用系统的性能和可扩展性，通过模拟大量用户并发执行关键任务，通过实时性能检测来确认问题和查找问题，并针对所发现的问题对系统性能进行优化。这类工具的代表有LoadRunner等。

使用工具Logiscope可以对程序进行静态结构分析，即不需要运行程序，仅通过语法扫描找出不符合编码规范之处，打印系统的调用关系图。

SmartBits软件提供了网络测试的功能。

（71）analysis emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately， the system models become the（72）for designing and constructing an improved system. （73）is such a technique. The emphasis in this technique is process-centered. Systems analysts draw a series of process models called （74）. （75）is another such technique that integrates data and process concerns into constructs called objects.

（71）A.Prototyping B.Accelerated C.Model-driven D.Iterative

（72）A.image B.picture C.layout D.blueprint

（73）A.Structured analysis B.Information Engineering

C.Discovery Prototyping D.Object-Oriented analysis

（75）A.PERT B.DFD C.ER D D.UML

（76）A.Structured analysis B.Information Engineering

C.Discovery Prototyping D.Object-Oriented analysis

**【答案】C D A B D**

**【解析】**

模型驱动的分析方法强调通过绘制图形化的系统模型来说明和验证已有的或待开发的系统。这些系统模型最终成为设计和构建改进后的系统的蓝图。结构化分析就是这样一种方法，它是以过程为中心的。在结构化分析方法中，系统分析员所绘制的一系列过程模型被称为DFD。面向对象分析是另外一种模型驱动的分析方法，它将数据和过程统一到了一个被称为“对象”的结构中。

**试题一**

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

**【说明】**

以下代码由 C 语言书写，能根据指定的年、月计算当月所含天数。

int GetMaxDay( int year， int month )

{

int maxday = 0;

if ( month >= 1 && month <= 12 )

{

if ( month == 2 )

{

if ( year % 4 == 0 )

{

if ( year % 100 == 0 )

{

if ( year % 400 == 0 )

maxday = 29;

else

maxday = 28;

}

else

maxday = 29;

}

else

maxday = 28;

}

else

{

if ( month == 4 || month == 6 || month == 9 || month == 11 )

maxday = 30;

else

maxday = 31;

}

}

return maxday;

}

**【问题1】**

请画出以上代码的控制流图。

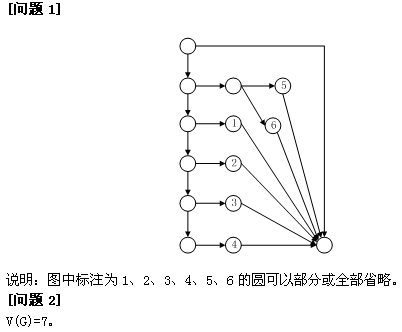
**【问题2】**

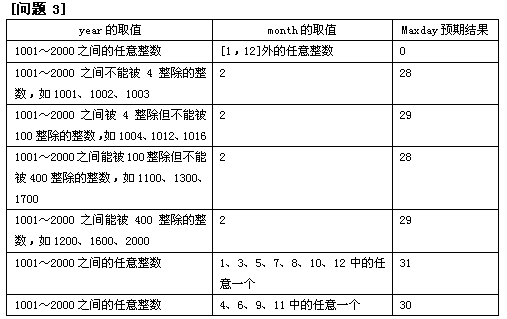
请计算上述控制流图的环路复杂度 V(G)。

**【问题 3】**

假设 year 的取值范围是 1000 < year < 2001，请使用基本路径测试法为变量 year、month 设计测试用例（写出 year 取值、month 取值、maxday 预期结果），使之满足基本路径覆盖要求。

**答案：**





**解析：**

本题考查白盒测试中的基本路径法的应用。

**【问题1】**

模块控制流图是与程序流程图相类似的由许多节点和连接节点的边组成的一种图形，其中一个节点代表一条语句，边表示节点间的控制流向，它显示了一个函数的内部逻辑结构。

**【问题2】**

V(G) = 区域数 = 判断节点数+1。

**【问题3】**

在进行程序的基本路径测试时，从程序的环路复杂性，可导出程序基本路径集合中的独立路径条数，这是确保程序中每个可执行语句至少执行1次所必须的测试用例数目的上界。独立路径是指包括一组以前没有处理的语句或条件的一条路径。

**试题二**

阅读下列说明，回答问题 1 至问题 5，将解答填入答题纸的对应栏内。

**【说明】**

负载压力性能测试是评估系统性能、性能故障诊断以及性能调优的有效手段。下述表格是针对税务征管系统中“税票录入”业务的测试结果，系统服务器端由应用服务器和单节点数据库服务器组成。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 并发用户数 | 交易吞吐量平均值（trans/s） | 交易响应时间平均值（s） | 数据库服务器CPU平均利用率 | 应用服务器CPU平均利用率 |
| 10 | 0.56 | 0.57 | 37.50% | 13.58% |
| 20 | 2.15 | 1.16 | 57.32% | 24.02% |
| 30 | 3.87 | 3.66 | 70.83% | 39.12% |
| 50 | 7.02 | 6.63 | 97.59% | 53.06% |

**【问题 1】（4 分）**

简述交易吞吐量和交易响应时间的概念。

**【问题 2】（2 分）**

试判断随着负载增加，当交易吞吐量不再递增时，交易响应时间是否会递增，并说明理由。

**【问题 3】（3 分）**

根据上述测试结果，判断服务器资源使用情况是否合理，为什么？

**【问题 4】（5 分）**

在并发用户数为 50 时，如果交易吞吐量和交易响应时间都不满足需求，简述数据库端造成此缺陷的主要原因，有效的解决方案是什么？

**【问题 5】（2 分）**

去年全年处理“税票录入”交易约100万笔，考虑到3年后交易量递增到每年200万笔。假设每年交易量集中在8个月，每个月20个工作日，每个工作日8小时，试采用 80/20 原理估算系统服务器高峰期 “税票录入”的交易吞吐量（trans/s）。

**答案：**

**【问题 1】**

交易吞吐量：系统服务器每秒能够处理通过的交易数。

交易响应时间：系统完成事务执行准备后所采集的时间戳和系统完成待执行事务后所采集的时间戳之间的时间间隔，是衡量特定类型应用事务性能的重要指标，标志了用户执行一项操作大致需要多长时间。

**【问题2】**

随着负载增加，当交易吞吐量不再递增时，交易平均响应时间一般会递增。当系统吞吐量达到极限时，客户端交易会在请求队列中排队等待，等待的时间会记录在响应时间中，故交易平均响应时间一般会递增。

**【问题 3】**

数据库服务器资源使用不合理。当用户数达到50时，数据库服务器CPU利用率达到97.5%，属于不合理范围。

**【问题 4】**

数据库端造成此缺陷的主要原因包括服务器资源负载过重、数据库设计不合理、数据库单个事务处理响应时间过长、系统并发负载造成最终用户响应时间过长。有效的解决方案是：采用数据库集群策略，并注意配置正确。

**【问题 5】**

（1000000\*2\*80%）/（8\*20\*8\*3600\*20%）=1.74trans/s。

**解析：**

本题考查负载压力性能测试的应用。

**【问题 1】**

考查交易吞吐量和交易响应时间的概念。

**【问题2】**

考查交易吞吐量与交易平均响应时间的关系。

当系统吞吐量未达到系统处理极限时，系统性能不会衰减，交易平均响应时间一般不会递增。当系统吞吐量达到极限时，客户端交易会在请求队列中排队等待，等待的时间会记录在响应时间中，故交易平均响应时间一般会递增。

**【问题 3】**

服务器资源使用中最主要的指标是CPU利用率，对于一般的应用系统，在负载压力期间，当CPU利用率平均值持续达到85%以上时，表示服务器资源使用出现瓶颈，视为不合理。

**【问题 4】**

当交易吞吐量和交易响应时间都不满足需求，数据库端造成此缺陷的主要原因，可以从数据库设计、资源使用以及应用开发几个角度去分析。最有效的解决方案是采用Oracle RAC集群策略。

**【问题5】**

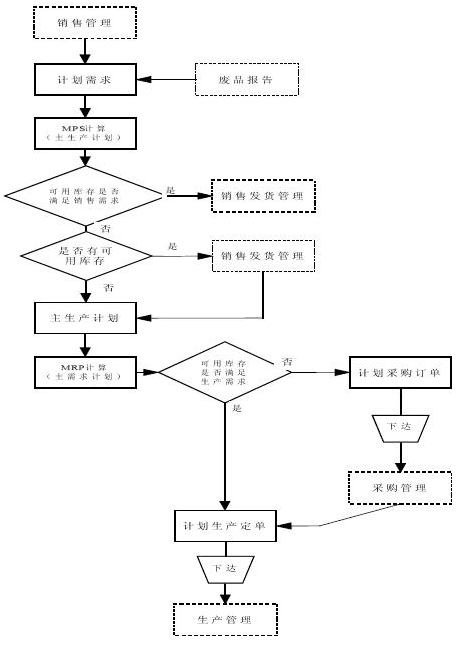
80/20法则的原理可以概括为：每个工作日中80%的业务在20%的时间内完成。本题中，业务量为1000000\*2，处理时间为8\*20\*8\*3600。

**试题三**

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

**【说明】**

在 ERP（企业资源计划）系统中，“计划管理”模块的流程图如下：



**【问题 1】**

因果图法是功能测试案例设计方法中的一种，简述利用因果图导出测试用例需要经过哪几个步骤？

**【问题 2】**

根据题干中的流程图，利用因果图法，找出哪些是正确的输入条件，那些是正确的输出结果，将下列描述的字母编号填入表中。

A.可以进行销售发货管理

B.可用库存不满足销售需求（有可用库存）

C.可用库存不满足销售需求（无可用库存）

D.可用库存满足生产需求

e.可用库存不满足生产需求

f.可用库存不满足生产需求（无可用库存）

g.可以进行 MPS 运算

h.可用库存满足销售需求

i.生成主生产计划

j.生成计划采购定单

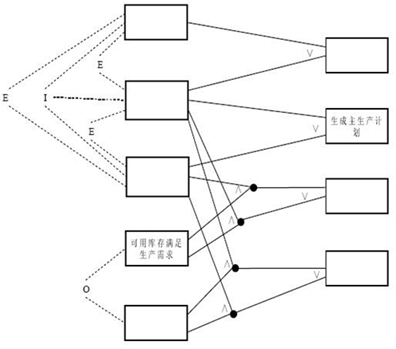
k.生成计划生产定单

l.可以进行 MRP 运算

|  |  |
| --- | --- |
| 输入条件 | 输出结果 |
|  |  |

**【问题 3】**

下图画出“计划管理”模块的因果图。请把问题 2 中列出的输入条件和输出结果的字母编号填入到空白框中相应的位置。



**答案：**

**【问题1】**

（1）分析程序规格说明的描述中，哪些是原因，哪些是结果。原因常常是输入条件或输入条件的等价类，而结果是输出条件。

（2）分析程序规格说明的描述中的语义的内容，并将其表示成连接各个原因与各个结果的因果图。

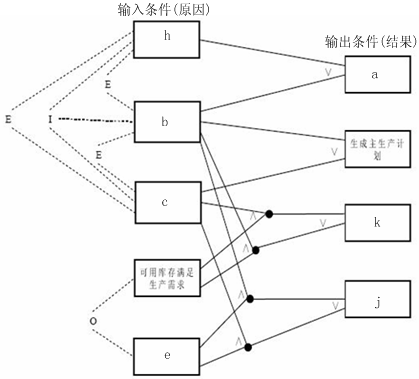
（3）标明约束条件。由于语法或环境的限制，有些原因和结果的组合情况不可能出现的。为标明这些特定的情况，在因果图上使用若干个标准的符号标明约束条件。

（4）把因果图转换成判定表，为判定表中每一列表示的情况设计测试用例。

**【问题2】**

输入条件：b、c、d、e、h；输出结果：a、i、j、k。

**【问题3】**



**解析：**

本题考查黑盒测试用例设计方法中的因果图法。

**【问题1】**

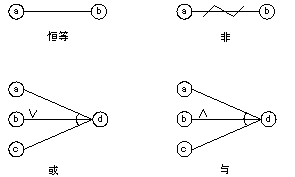
因果图法的思路是：从用自然语言书写的程序规格说明的描述中找出因（输入条件）和果（输出或程序状态的改变），通过因果图转换为判定表。

**【问题2】**

依据测试人员对ERP业务的理解，可知：正确的输入条件中不包括可用库存不满足生产需求（无可用库存）；正确的输出结果中不包括可以进行MPS运算和可以进行MRP运算。

**【问题3】**

正确生成因果图，第一需要测试人员掌握测试业务的逻辑关系，第二需要测试人员掌握因果图的绘制方法。因果图中的基本图形符号和约束符号说明如下：



图中左边的节点表示原因，右边的节点表示结果。恒等、非、或、与的含义如下：

（1）恒等：若a=1，则b=1；若a=0，则b=0。

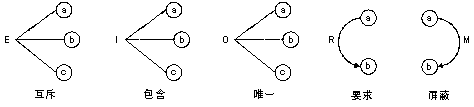
（2）非：若a=1，则b=0；若a=0，则b=1。

（3）或：若a=1或b=1或c=1，则d=1；若a=b=c=0，则d=0。

（4）与：若a=b=c=1，则d=1；若a=0或b=0或c=0，则d=0。

画因果图时，原因在左，结果在右，由上到下排列，并根据功能说明书中规定的原因和结果之间的关系，用上述基本符号连接起来。在因果图中还可以引入一些中间节点。

由于语法或环境限制，有些原因与原因之间、原因与结果之间的组合情况不可能出现。为表明这些特殊情况，在因果图上用一些记号表明约束或限制条件。因果图的约束条件如下图所示。



其中互斥、包含、唯一、要求是对原因的约束，屏蔽是对结果的约束。它们的含义如下：

（1）互斥：表示不同时为1，即a，b，c中至多只有一个1。

（2）包含：表示至少有一个1，即a，b，c中不同时为0。

（3）唯一：表示a，b，c中有且仅有一个1。

（4）要求：表示若a=1，则b必须为1。即不可能a=1且b=0。

（5）屏蔽：表示若a=1，则b必须为0。

**试题四**

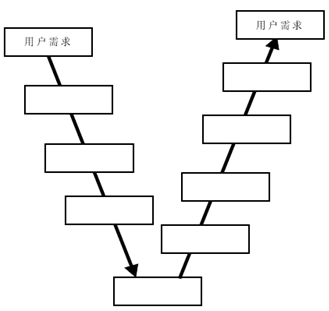
阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

**【说明】**

软件测试是与软件开发紧密相关的一系列有计划的系统性活动。软件测试需要用测试模型去指导实践，软件测试专家通过测试实践总结出了很多很好的测试模型。

**【问题 1】**

V模型是最具有代表意义的软件测试模型。请将开发活动和相应的测试活动填入下图空白框中。



**【问题 2】**

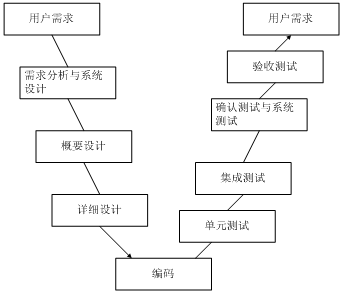
在集成测试阶段，可采用不同的组装方式把模块组装起来形成一个可运行的系统，其中增殖式组装方式包括哪几种？除增殖式组装方式外还有哪种组装方式？

**【问题 3】**

测试工程师甲按照V模型安排测试活动，在验收测试阶段发现的某些功能缺陷是与产品需求设计说明书有关，造成软件缺陷修复成本较高。你认为若采用W模型能否避免类似问题出现？简述W模型的优点。

**答案：**

**【问题1】**



**【问题2】**

（1）自顶向下的增殖方式、自底向上的增殖方、混合增殖方式。

（2）除了增殖方式外，还有一次性组装方式。

**【问题3】**

能

（1）测试的活动应该与软件开发同步进行。

（2）测试的对象不仅仅是程序，还包括需求和设计。

（3）尽早发现软件缺陷可降低软件开发的成本。

**解析：**

本题考查测试模型的应用。

**【问题1】**

考查V模型的概念。

**【问题2】**

确认测试与系统测试阶段的测试内容包括：

（1）功能测试：目标中的功能是否真实实现了。

（2）安全性测试：设法破坏程序的安全性、保密检查。

（3）可靠性测试：平均无故障时间、因故障停机时间，需要模拟运行环境。

（4）可扩充性测试：指系统软件与硬件的可扩展能力。

（5）兼容性测试：向下兼容、交错兼容。

（6）易用性测试：在指定条件下使用时，软件产品被理解、学习、使用和吸引用户的能力。

**【问题3】**

V模型的主要缺陷是在验收测试阶段发现的某些功能缺陷是与产品需求设计说明书有关，造成软件缺陷修复成本较高。W模型对V模型的改进主要有三个方面：

（1）测试的活动应该与软件开发同步进行。

（2）测试的对象不仅仅是程序，还包括需求和设计。

（3）尽早发现软件缺陷可降低软件开发的成本。

**试题五**

阅读以下说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

**【说明】**

企业在质量方面的投资会产生利润，诸如提高产品质量会提高公司的声誉，降低产品交付之后的维护成本等等。测试是重要的质量保证手段，但必须付出相应的测试成本。

**【问题 1】（3 分）**

简述测试实施成本的构成。

**【问题 2】（3 分）**

简述缺陷探测率的计算公式。

**【问题 3】（8 分）**

假设对一个开发的 MIS（管理信息系统）系统进行测试。属于质量预防方面的一致性成本只考虑软件测试的投资，把发布之前和之后发现修改缺陷产生的成本看成非一致性成本。假设发现的缺陷为 300 个，各阶段花费在发现及修改缺陷的成本假设如下：

在开发过程单元测试阶段，软件开发人员发现及修改每一个缺陷的成本为 100元；在独立测试阶段进行集成和系统测试，测试人员发现开发人员修改、测试人员再确认每一个缺陷的成本为 400元；在产品发布后，由客户发现，报告技术支持人员、相关开发人员修改，测试组再进行回归测试，每一个缺陷的成本为 4000元。

请对比这3 种测试情况，填补下表空格中的内容。

|  |  |  |
| --- | --- | --- |
| **质量成本项** | **测试成本项** | **自动测试** |
| 测试投资 | 测试人工费 | 50000 |
| 环境使用费 | 10000 |
| 测试工具费 | 15000 |
| 测试总投资 |  |
| 单元测试 | 发现缺陷数 | 80 |
| 每个缺陷成本 | 100 |
| 内部（开发）缺陷成本 |  |
| 独立测试 | 发现缺陷数 | 215 |
| 每个缺陷成本 | 400 |
| 内部（测试）缺陷成本 |  |
| 回归测试 | 发现缺陷数 | 5 |
| 每个缺陷成本 | 4000 |
| 内部缺陷成本 |  |
| 质量成本 | 一致性成本 |  |
| 非一致性成本 |  |
| 总质量成本 |  |
| DDP | 缺陷探测率 |  |

**答案：**

**【问题1】**

测试准备成本、测试执行成本、测试结束成本。

**【问题2】**

缺陷探测率 = 测试者发现的错误数/（测试者发现的错误数+客户发现并反馈给技术支持人员进行修复的错误数）

**【问题3】**



**解析：**

总质量成本中包括一致性成本和非一致性成本。

一致性成本是指用于保证软件质量的支出，包括预防性成本和测试预算，如测试计划、测试开发和测试实施费用等。测试预算被称为审查费。

非一致性成本是由出现的软件错误和测试过程故障（如延期、劣质的测试发布）引起的。这些问题会导致测试返工、补测、延迟。追加测试时间和资金就是一种由于内部故障引起的非一致性成本。非一致性成本还包括外部故障（软件遗留错误影响客户）引起部分。这些成本还包括技术支持小组预算、错误修正花费、产品收回、赔偿和销售成本。

缺陷探测率 = 测试者发现的错误数/（测试者发现的错误数+客户发现并反馈给技术支持人员进行修复的错误数）