

NEW PLAN

软考高级架构师

• — 段 新 征 程 •





大纲介绍

项目管理

面向对象开发

软件工程

软件工程概述

软件成熟度模型

软件过程模型

逆向工程

需求工程

系统设计

测试基础知识

系统运行与维护

净室软件工程

基于构件的软件工程



测试原则



系统测试： 为了发现错误而执行程序的过程，成功的测试是发现了至今尚未发现的错误的测试。



测试原则：

应尽早并不断的进行测试，比如V模型，从设计的时候就开始测试；

- 测试工作应该避免由原开发软件的人或小组承担；
- 在设计测试方案时，不仅要确定输入数据，而且要根据系统功能确定预期的输出结果；
- 既包含有效、合理的测试用例，也包含不合理、失效的用例；
- 检验程序是否做了该做的事，且是否做了不该做的事；
- 严格按照测试计划进行；
- 妥善保存测试计划和测试用例；
- 测试用例可以重复使用或追加测试。



测试方法



软件测试方法可分为**静态测试**和**动态测试**。



静态测试：指被测试程序不在机器上运行，而采用**人工检测**和**计算机辅助静态分析**的手段对程序进行检测，包括对文档的静态测试和对代码的静态测试。包括**桌前检查**、**代码审查**、**代码走查**的方式。使用这种方法能够有效地发现**30%-70%**的逻辑设计和编码错误。

- **桌前检查**：在你自己编写代码后，你会进行桌前检查，这意味着你会仔细查看你的代码，以捕捉可能的错误、逻辑问题或风格不一致之类的问题。这是一种个人层面的检查，有助于在代码提交前修复问题。
- **代码审查**：代码审查是团队中的成员一起对某个人编写的代码进行检查。通过代码审查，团队可以共同评估代码的质量，发现潜在的问题，并确保代码符合团队的标准和最佳实践。这有助于提高代码的稳定性和可维护性。
- **代码走查**：代码走查是一种更广泛的审查实践，通常包括团队的开发人员、测试人员和其他相关人员。在代码走查过程中，团队会深入检查代码的各个方面，包括逻辑、性能、安全性等。目标是确保代码在整体上是健壮、高效且符合预期要求的。



测试方法



动态测试：指在计算机上**实际运行程序**进行软件测试，一般采用白盒测试和黑盒测试方法(还有灰盒和自动化)。

- **黑盒测试：**黑盒测试关注于测试**软件的功能(功能性测试)**，而不考虑内部实现细节。测试人员不需要知道代码的具体结构，而是根据软件的需求规格和功能来设计测试用例。这种方法类似于将测试人员置于一个“盒子”外，只观察软件的输入和输出，以确定是否按预期工作。
 - **举例：**假设你正在测试一个在线登录系统。对于黑盒测试，你会设计测试用例，包括输入不同的用户名和密码组合，然后观察系统的响应，验证是否成功登录、失败登录是否有适当的错误提示等，而不考虑系统内部的代码结构。
- **白盒测试：**白盒测试关注于测试软件的内部逻辑和**代码结构(结构性测试)**，以确保代码按照预期方式执行。测试人员需要了解软件的代码，以设计测试用例，以覆盖不同的代码路径和分支情况，以及验证代码是否满足质量标准和最佳实践。
 - **举例：**假设你正在测试一个计算器应用。对于白盒测试，你会检查代码，确保加法、减法、乘法和除法等操作都正确实现。你可能会编写测试用例，测试各种输入情况，例如测试正数、负数、小数等，以确保代码在不同情况下都能正确执行。



测试阶段



1、单元测试：也称为**模块测试**，测试的对象是可**独立编译或汇编的程序模块、软件构件或OO软件中的类(统称为模块)**，测试依据是**软件详细设计说明书**。



2、集成测试：目的是**检查模块之间，以及模块和已集成的软件之间的接口关系**，并验证已集成的软件是否符合设计要求。测试依据是**软件概要设计文档**。



3、系统测试：**测试对象是完整的、集成的计算机系统**；测试的目的是在真实系统工作环境下，验证完成的软件配置项能否和系统正确连接，并满足系统/子系统设计文档和软件开发合同规定的要求。**测试依据是用户需求或开发合同**。主要内容包括功能测试、健壮性测试、性能测试、用户界面测试、安全性测试、安装与反安装测试等，其中，**最重要的工作是进行功能测试与性能测试**。功能测试主要采用黑盒测试方法；性能测试主要指标有响应时间、吞吐量、并发用户数和资源利用率等。系统测试通常由独立的测试团队执行，他们并不直接参与软件的开发过程。



测试阶段



4、确认测试：主要用于验证软件的功能、性能和其他特性是否与用户需求一致。测试依据是需求文档，确认测试是软件或产品开发的最后一个阶段，在系统测试完成后进行。它的主要目标是确保软件或产品已经满足最终用户的期望和需求。在确认测试中，最终用户（或其代表）将根据他们的实际使用情境，验证软件是否符合他们的业务流程和预期目标根据用户的参与程度，通常包括以下类型：

- 内部确认测试：主要由软件开发组织内部按照SRS进行测试。
- Alpha测试：用户在开发环境下进行测试。
- Beta测试：用户在实际使用环境下进行测试，通过改测试后，产品才能交付用户。
- 验收测试：针对SRS,在交付前以用户为主进行的测试。其测试对象为完整的、集成的计算机系统。验收测试的目的是，在真实的用户工作环境下，检验软件系统是否满足开发技术合同或SRS。验收测试的结论是用户确定是否接收该软件的主要依据。除应满足一般测试的准入条件外，在进行验收测试之前，应确认被测软件系统已通过系统测试。



5、回归测试：测试目的是测试软件变更之后，变更部分的正确性和对变更需求的符合性，以及软件原有的、正确的功能、性能和其他规定的要求的不损害性(不能把其他好的模块改错)。



考试真题



软件确认测试也称为有效性测试，主要验证()。确认测试计划通常是在需求分析阶段完成的。根据用户的参与程度不同，软件确认测试通常包括()。

- A.系统中各个单元模块之间的协作性
- B.软件与硬件在实际运行环境中能否有效集成
- C.软件功能、性能及其它特性是否与用户需求一致
- D.程序模块能否正确实现详细设计说明中的功能、性能和设计约束等要求

A.黑盒测试和白盒测试

B.一次性组装测试和增量式组装测试

C.内部测试、Alpha、Beta和验收测试

D.功能测试、性能测试、用户界面测试和安全性测试



软件测试一般分为两个大类：动态测试和静态测试。前者通过运行程序发现错误，包括()等方法；后者采用人工和计算机辅助静态分析的手段对程序进行检测，包括()等方法。

A.边界值分析、逻辑覆盖、基本路径

B.桌面检查、逻辑覆盖、错误推测

C.桌面检查、代码审查、代码走查

D.错误推测、代码审查、基本路径

A.边界值分析、逻辑覆盖、基本路径

B.桌面检查、逻辑覆盖、错误推

C.桌面检查、代码审查、代码走查

D.错误推测、代码审查、基本路径



测试用例的设计

黑盒测试用例：将程序看做一个黑盒子，只知道输入输出，不知道内部代码，由此设计出测试用例，分为下面几类：



- **等价类划分：**等价类划分是一种方法，通过将输入值分成不同的类别，从而减少测试用例的数量，同时确保测试覆盖各种可能的情况。这样可以更高效地测试软件，而不必为每个可能的输入值都编写一个测试用例。



- **举例：**假设你在测试一个在线订餐系统，用户需要输入送餐地址的街道。等价类划分可以帮助你更有效地设计测试用例。

等价类1：合法的街道名，如 "Main Street" 或 "Maple Avenue"。

等价类2：数字街道名，如 "123rd Street"。

等价类3：特殊字符，如 "!" 或 "%"。

等价类4：空白输入。

在这个例子中，你可以为每个等价类设计一个测试用例，以确保系统能够正确处理不同类型的输入情况。例如，你可以测试一个合法的街道名来确保订单可以成功提交，同时测试一个数字街道名和特殊字符，以验证系统是否能够正确拒绝无效的输入。这样，通过代表不同等价类的测试用例，你可以有效地覆盖各种可能的情况。



- **边界值划分：**将每类的边界值作为测试用例，边界值一般为范围的两端值以及在此范围之外的与此范围间隔最小的两个值。例如，范围为0-150，边界值为0,150,-1,151四个。



- **错误推测：**没有固定的方法，凭经验而言，来推测有可能产生问题的地方，作为测试用例进行测试。



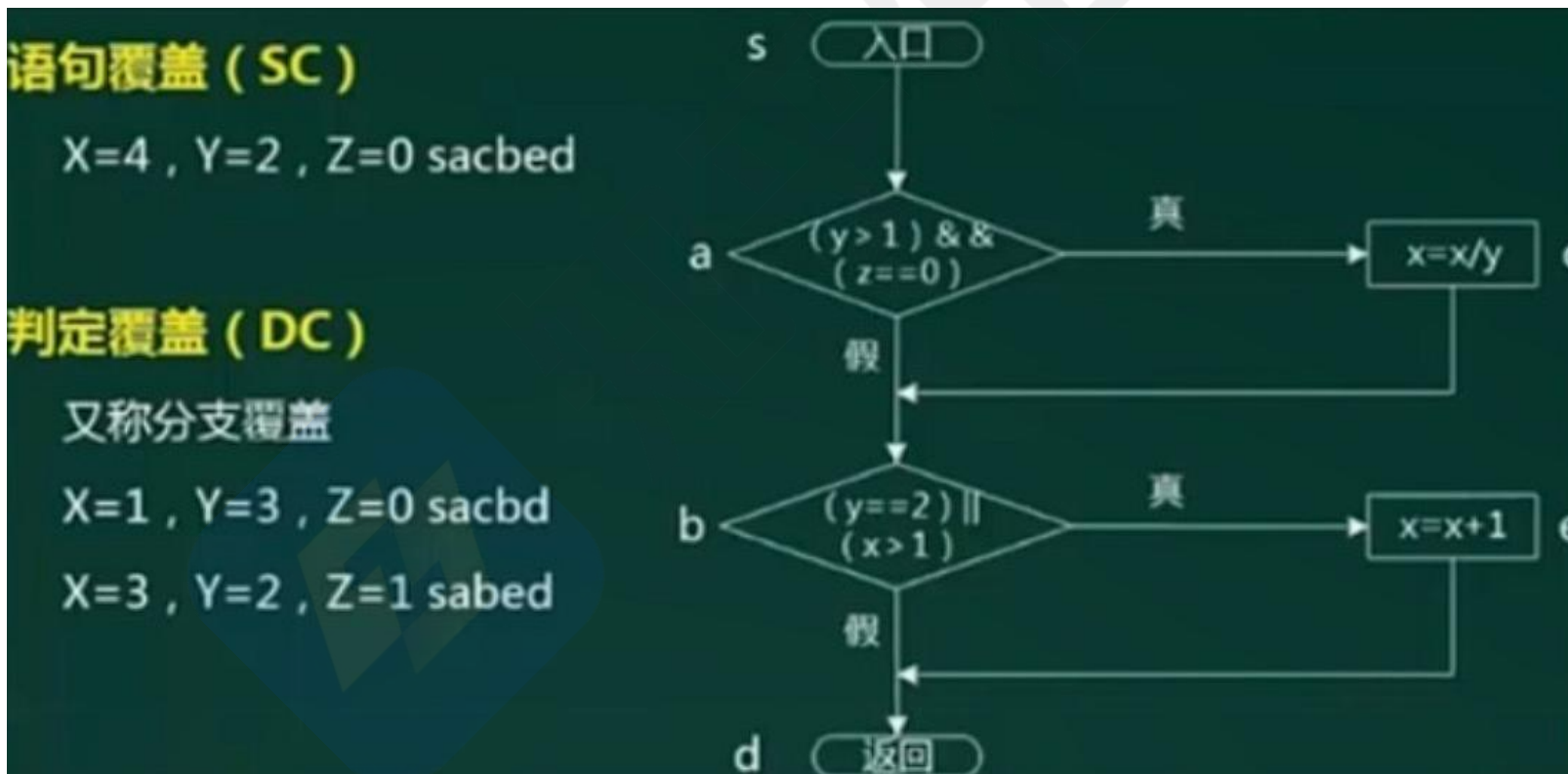
- **因果图：**由一个结果来反推原因的方法，具体结果具体分析，没有固定方法。



测试用例的设计

👁 黑盒白盒测试用例：知道程序的代码逻辑，按照程序的代码语句，来设计覆盖代码分支的测试用例，覆盖级别从低至高分为下面几种：

- 语句覆盖SC:逻辑代码中的所有语句都要被执行一遍，覆盖层级最低，因为执行了所有的语句，不代表执行了所有的条件判断。
- 判定覆盖DC:逻辑代码中的所有判断语句的条件的真假分支都要覆盖一次。





测试用例的设计

- 条件覆盖CC: 针对每一个判断条件内(比如一个if)的每一个独立条件(if中的每个判定条件)都要执行一遍真和假。
- 条件判定组合覆盖CDC: 同时满足判定覆盖和条件覆盖。

条件覆盖 (CC)

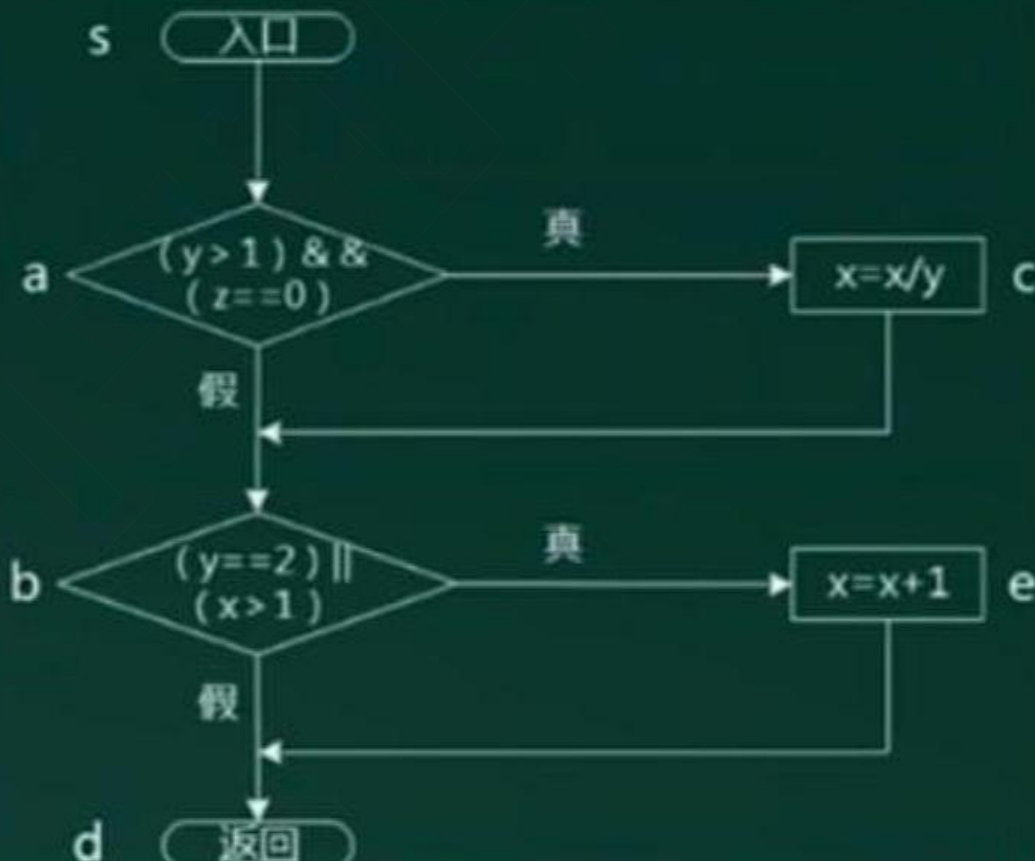
X=1, Y=2, Z=0 saced

X=2, Y=1, Z=1 sabed

条件判定组合覆盖 (CDC)

X=4, Y=2, Z=0 saced

X=1, Y=1, Z=1 sabd



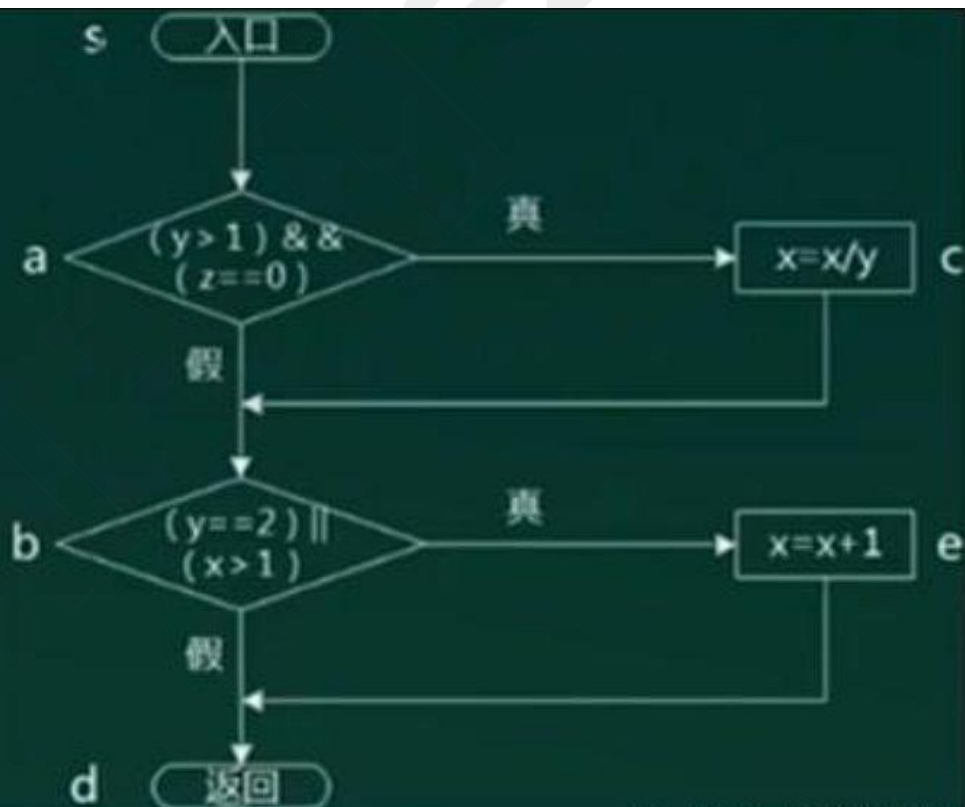


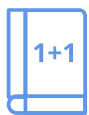
测试用例的设计

- 路径覆盖：逻辑代码中的所有可行路径都覆盖了，覆盖层级最高。

路径覆盖

X=1, Y=1, Z=1	sabd
X=3, Y=2, Z=0	sacbed
X=3, Y=3, Z=0	sacbd
X=1, Y=2, Z=1	sabed





招聘系统要求求职的人年龄在20岁到60岁之间(含),学历为本科、硕士或者博士,专业为计算机科学与技术、通信工程或者电子工程。其中()不是好的测试用例。

- A.(20,本科, 电子工程) B.(18,本科, 通信工程)
- B.(18,大专, 电子工程) D.(25,硕士, 生物学)



以下关于测试的叙述中, 正确的是()。

- A.实际上, 可以采用穷举测试来发现软件中的所有错误
- B.错误很多的程序段在修改后错误一般会非常少
- C.测试可以用来证明软件没有错误
- D.白盒测试技术中, 路径覆盖法往往能比语句覆盖法发现更多的错误



考试真题



采用白盒测试方法对下图进行测试，设计了4个测试用例：①($x=0, y=3$), ②($x=1, y=2$), ③($x=-1, y=2$), ④($x=3, y=1$)。至少需要测试用例①②才能完成(35)覆盖，至少需要测试用例①②③或①②④才能完成(36)覆盖。

A. 语句

B. 条件

C. 判定

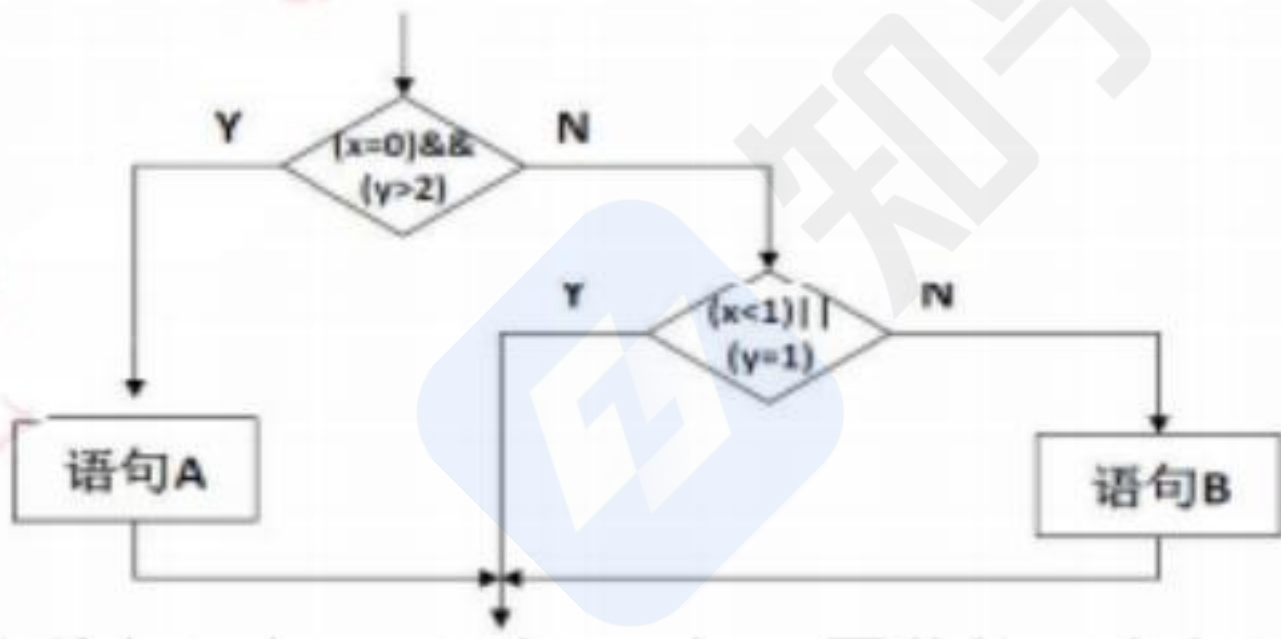
D. 路径

A. 语句

B. 条件

C. 判定

D. 路径



答案：A D

解析：可根据问题将测试用例代入执行，①②会执行语句A和语句B,但只覆盖第一个条件，因此是语句覆盖；①②覆盖了第一个条件的真假，执行了第一个条件的Y和N以及第二个条件的N,而③或者④又执行到了第二个条件的Y,因此流程中所有分支都走到了，是路径覆盖。



调试



测试是发现错误，调试是找出错误的代码和原因。



调试需要确定错误的准确位置；确定问题的原因并设法改正；改正后要进行回归测试。



调试的方法有：蛮力法、回溯法(从出错的地方开始，向回找)、原因排除法(找出所有可能的原因，逐一进行排除)。



软件的两种属性：外部属性指面向管理者和用户的属性，可直接测量，一般为性能指标。内部属性指软件产品本身的属性，如可靠性等，只能间接测量。



McCabe度量法：又称为环路复杂度，假设有向图中有向边数为 m ,节点数为 n ,则此有向图的环路复杂度为 $m-n+2$ 。



注意 m 和 n 代表的含义不能混淆，可以用一个最简单的环路来做特殊值记忆此公式，另外，针对一个程序流程图，每一个分支边(连线)就是一条有向边，每一条语句(语句框)就是一个顶点。



考试真题



下图用白盒测试方法进行测试，图中有()条路径，采用McCabe度量计算该程序图的环路复杂性为()

A、3

B、4

C、5

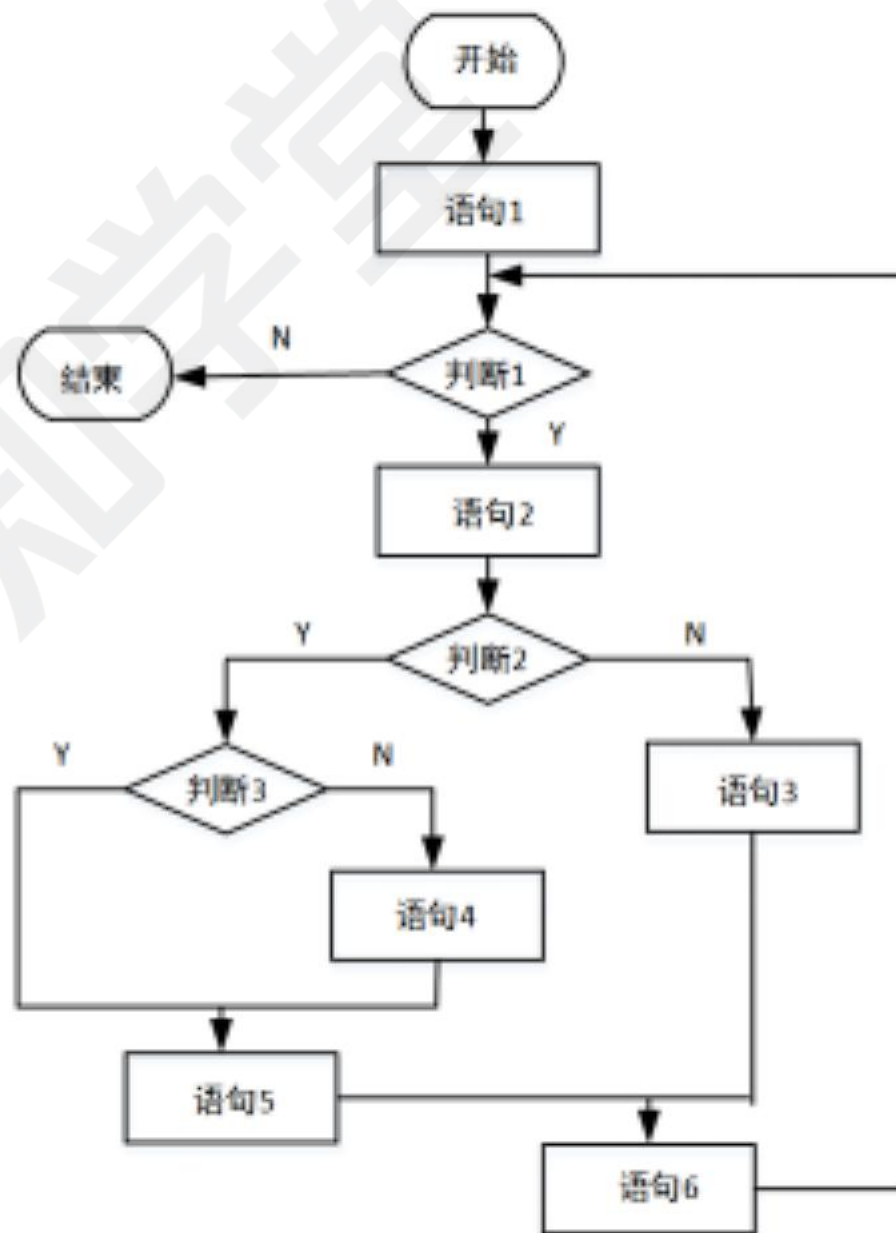
D、6

A、3

B、4

C、5

D、6



T H E E N D

功不唐捐，玉汝于成！

• 开 启 新 征 程 •

