全国计算机技术与软件专业技术资格（水平）考试最实用真题用书

全国计算机技术与软件专业技术资格(水平)考试

历年真题必练

（含关键考点点评）

—程序员

研究历年真题是加分致胜的法宝
掌握核心考点是考试过关的关键

全国计算机专业技术资格考试真题研究组　编写

北京邮电大学出版社
www.buptpress.com

历年真题必练

（含关键考点点评）

一风作品

全国计算机技术与软件专业技术资格（水平）考试历年真题必练

（含关键考点点评）

——程序员

全国计算机专业技术资格考试真题研究组　编写

北京邮电大学出版社

·北京·

内 容 简 介

本书以最新版的计算机技术与软件专业技术资格(水平)考试程序员考试大纲为指导,包括最新8套全真试题(上、下午)+试题详细解析+关键考点评注。8套全真试题,给考生提供8次实战演练机会。特别需要指出的是,本书每套试卷后均配有关键考点评注,方便考生快速重温重点难点,迅速提高应试能力。特别地,本书在深入研究历年真题的基础上,梳理归类出同源考点真题,总结命题规律,指引命题方向。

本书可供全国计算机技术与软件专业技术资格(水平)考试程序员考生复习使用,特别适合考前冲刺使用,同时也可作为相关培训班的教材。

图书在版编目(CIP)数据

全国计算机技术与软件专业技术资格(水平)考试历年真题必练. 程序员/全国计算机专业技术资格考试真题研究组编写.--北京:北京邮电大学出版社,2013.9

含关键考点点评

ISBN 978-7-5635-3642-9

Ⅰ.①全… Ⅱ.①全… Ⅲ.①程序设计—工程技术人员—资格考试—习题集 Ⅳ.①TP3-44

中国版本图书馆CIP数据核字(2013)第189955号

书　　名:全国计算机技术与软件专业技术资格(水平)考试历年真题必练(含关键考点点评)——程序员

作　　者:全国计算机专业技术资格考试真题研究组

责任编辑:满志文

出版发行:北京邮电大学出版社

社　　址:北京市海淀区西土城路10号(100876)

发 行 部:电话:010-62282185　传真:010-62283578

E-mail:publish@bupt.edu.cn

经　　销:各地新华书店

印　　刷:北京联兴华印刷厂

开　　本:787 mm×1092 mm　1/16

印　　张:11.5

字　　数:423千字

版　　次:2013年9月第1版　2013年9月第1次印刷

ISBN 978-7-5635-3642-9　　定价:27.00元

前　言

全国计算机技术与软件专业技术资格(水平)考试(以下简称计算机软件考试)是由国家人力资源和社会保障部、工业和信息化部领导下的国家级考试,其目的是,科学、公正地对全国计算机与软件专业技术人员进行职业资格、专业技术资格认定和专业技术水平测试。该考试由于其权威性和严肃性,得到了社会及用人单位的广泛认同,并为推动我国信息产业特别是软件产业的发展和提高各类IT人才的素质做出了积极的贡献。

全国计算机软件考试是一种水平性考试,历年真题具有极强的规律性和重复性,通过研究我们发现一个惊人的事实:几乎每年都有2—3题是以前考过的真题,约有72%是雷同的考点,有变化的新考题仅仅有约9%!也就是说,只要把考过的真题都会做,就能轻松获过关。为了帮助准备参加计算机软件考试的应试者更好地复习迎考,我们组织编写了这套《全国计算机技术与软件专业技术资格(水平)考试历年真题必练》丛书。

本丛书突出如下特点:

(1)真题套数多。本书包括最新8套全真试题(上、下午)+试题详细解析+关键考点评注,供考生全面复习与突破过关。

(2)答案解析,详略得当:试卷不仅给出了参考答案,且一一予以解题分析,突出重点、难点,详略得当,力求通过解析的学习,强化理解、记忆。

(3)每套试题解析最后附有关键考点评注。同类图书一般是"试卷+解析"的风格,我们根据培训老师的实际培训经验,在每套试卷解析最后加了"关键考点评注",对本套试卷中难点、重点进行剖析,使考生能达到举一反三功效;对重点考点进行链接,使考生重温了相关知识点,备考更有信心。

(4)真题归类研究,把握命题规律。本书在深入研究历年真题的基础上,梳理归类出同源考点真题,总结命题规律,指引命题方向。

(5)装帧特独,便于自测。每套试题按"试卷+解析+评注"装成一份,非常适合考生每份试题按"练、学、查"方式实战,而且充分考虑到培训班的特点,方便教学使用。

(6)作者实力强。作者团队系从事计算机软件考试近10年的辅导、培训、命题、阅卷及编写之经验,有较高的权威性,图书质量有保障。

本书可供全国计算机技术与软件专业技术资格(水平)考试网络工程师考生复习使用,特别适合考前冲刺使用,同时也可作为相关培训班的教材。

本书由全国软考新大纲命题研究组主编,参与编写的人员有:张源源、董自涛、牛雪飞、王芳、周汉、高玲云、朱恽、汤小燕、刘志强、钟彩华、张天云、任培花、王莉、朱世昕、赵鹏、孙玫、杨剑、王玉玺、曹愚、刘鹏、何光明等。在本书编写过程中,参考了许多相关的书籍和资料,编者在此对这些参考文献的作者表示感谢。

因作者水平有限,书中难免存在错漏和不妥之处,望读者批评指正,联系邮箱:iteditor@126.com。

编　者

目　录

2013 年 5 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 22 页）

上午试卷 …… 1
下午试卷 …… 6
上午试卷答案及解析 …… 13
下午试卷答案及解析 …… 19
关键考点点评 …… 20

2012 年 11 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 21 页）

上午试卷 …… 1
下午试卷 …… 7
上午试卷答案及解析 …… 13
下午试卷答案及解析 …… 19
关键考点点评 …… 20

2012 年 5 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 22 页）

上午试卷 …… 1
下午试卷 …… 7
上午试卷答案及解析 …… 14
下午试卷答案及解析 …… 19
关键考点点评 …… 21

2011 年 11 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 26 页）

上午试卷 …… 1
下午试卷 …… 7
上午试卷答案及解析 …… 15
下午试卷答案及解析 …… 22
关键考点点评 …… 24

2011 年 5 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 24 页）

上午试卷 …… 1
下午试卷 …… 7
上午试卷答案及解析 …… 14
下午试卷答案及解析 …… 20
关键考点点评 …… 22

2010 年 11 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 21 页）

上午试卷 …… 1
下午试卷 …… 7
上午试卷答案及解析 …… 14
下午试卷答案及解析 …… 18
关键考点点评 …… 20

2010 年 5 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 23 页）

上午试卷 …… 1
下午试卷 …… 8
上午试卷答案及解析 …… 15
下午试卷答案及解析 …… 20
关键考点点评 …… 21

2009 年 11 月全国计算机技术与软件专业技术资格(水平)考试程序员（共 18 页）

上午试卷 …… 1
下午试卷 …… 7
上午试卷答案及解析 …… 13
下午试卷答案及解析 …… 16
关键考点点评 …… 17

2013年5月全国计算机技术与软件专业技术资格(水平)考试程序员

上午试卷

(考试时间 150 分钟,满分 75 分)

本试卷的试题中共有 75 个空格,需要全部解答,每个空格 1 分,满分 75 分。每个空格对应一个序号,有A、B、C、D 四个选项,请选择一个最恰当的选项作为解答,在答题卡相应序号下填涂该选项。

• 在 Word 的编辑状态下,若要防止在段落中间出现分页符,可以通过单击鼠标右键在弹出的菜单中选择__(1)__命令;在"段落"对话框中,选择"换行和分页"选项卡,然后再勾选__(2)__。

(1)A. 段落(P)…　　B. 插入符号(S)　　C. 项目符号(B)　　D. 编号(N)

(2)A. 孤行控制　　B. 与下段同页

C. 段中不分页　　D. 段前分页

• 某 Excel 工作表如下所示,若在 D1 单元格中输入=$A$1+$B$1+C1,则 D1 的值为__(3)__;此时,如果向垂直方向拖动填充柄至 D3 单元格,则 D2 和 D3 的值分别为__(4)__。

	A	B	C	D
1	16	18	20	
2	23	26	30	
3	35	38	26	
4				

(3)A. 34　　B. 36　　C. 39　　D. 54

(4)A. 79 和 99　　B. 69 和 93　　C. 64 和 60　　D. 79 和 93

• __(5)__服务的主要作用是实现文件的上传和下载。

(5)A. Gopher　　B. FTP　　C. Telnet　　D. E-mail

• 与八进制数 1706 等值的十六进制数是__(6)__。

(6)A. 3C6　　B. 8C6　　C. F18　　D. F1C

• 若计算机字长为 8,则采用原码表示的整数范围为−127～127,其中,__(7)__占用了两个编码。

(7)A. −127　　B. 127　　C. −1　　D. 0

• CPU 执行指令时,先要根据__(8)__将指令从内存读取出并送入__(9)__,然后译码并执行。

(8)A. 程序计数器　　B. 指令寄存器

C. 通用寄存器　　D. 索引寄存器

(9)A. 程序计数器　　B. 指令寄存器

C. 地址寄存器　　D. 数据寄存器

• 显示器的性能指标主要包括__(10)__和刷新频率。若显示器的__(11)__,则图像显示越清晰。

(10)A. 重量　　B. 分辨率　　C. 体积　　D. 采样速度

(11)A. 采样频率越高　　B. 体积越大　　C. 分辨率越高　　D. 重量越重

• 图像文件格式分为静态图像文件格式和动态图像文件格式。__(12)__属于静态图像文件格式。

(12)A. MPG　　B. AVS　　C. JPG　　D. AVI

• 将声音信号数字化时，__(13)__不会影响数字音频数据量。

(13)A. 采样率　　B. 量化精度　　C. 波形编码　　D. 音量放大倍数

• 计算机系统中，内存和光盘属于__(14)__。

(14)A. 感觉媒体　　B. 存储媒体　　C. 传输媒体　　D. 显示媒体

• 对计算机软件的法律保护不涉及__(15)__。

(15)A. 知识产权法　　B. 著作权法　　C. 刑法　　D. 合同法

• 以下知识产权保护对象中，__(16)__不具有公开性基本特征。

(16)A. 科学作品　　B. 发明创造　　C. 注册商标　　D. 商业秘密

• 防火墙的 NAT 功能主要目的是__(17)__。

(17)A. 进行入侵检测　　B. 隐藏内部网络 IP 地址及拓扑结构信息

C. 防止病毒入侵　　D. 对应用层进行侦测和扫描

• 脚本漏洞主要攻击的是__(17)__。

(18)A. PC　　B. 服务器　　C. 平板电脑　　D. 智能手机

• 工作时需要动态刷新的是__(18)__。

(19)A. DRAM　　B. PROM　　C. EPROM　　D. SRAM

• 若计算机字长为 64 位，则用补码表示时的最小整数为__(19)__。

(20)A. -2^{64}　　B. -2^{63}　　C. $-2^{64}+1$　　D. $-2^{63}+1$

• 对于容量为 32K×32 位、按字编址(字长为 32)的存储器，其地址线的位数应为__(20)__。

(21)A. 15　　B. 32　　C. 64　　D. 5

• 对于一个值不为 0 的整数 x，进行__(22)__运算后结果为 0。

(22)A. x 与 x 按位与　　B. 将 x 按位取反

C. x 与 x 按位或　　D. x 与 x 按位异或

• 在操作系统设备管理中，通常不能采用__(23)__分配算法。

(23)A. 先来先服务　　B. 时间片轮转　　C. 单队列优先　　D. 多队列优先

• Windows 磁盘碎片整理程序__(23)__，通过对磁盘进行碎片整理，__(24)__。

(24)A. 只能将磁盘上的可用空间合并为连续的区域

B. 只能使每个操作系统文件占用磁盘上连续的空间

C. 可以使每个文件和文件夹占用磁盘上连续的空间，合并盘上的可用空间

D. 可以清理磁盘长期不用的文件，回收并占用空间使其成为连续的区域

(25)A. 可以提高对文件和文件夹的访问效率

B. 只能提高对文件夹的访问效率，但对文件的访问效率保持不变

C. 只能提高系统对文件的访问效率，但对文件夹的访问效率保持不变

D. 可以将磁盘空间的位示图管理方法改变为空闲区管理方法

• 在段页式管理中，如果地址长度为 32 位，并且地址划分如下图所示：

10 位	10 位	12 位
段号	页号	页内地址

在这种情况下，系统页面的大小应为__(26)__KB，且__(27)__。

(26)A. 1　　B. 2　　C. 3　　D. 4

(27)A. 最少有 1024 个段，每段最大为 4096KB

B. 最多有 1024 个段，每段最大为 4096KB

C. 最少有 1024 个段，每段最小为 4096KB

D. 最多有 1000 个段，每段最小为 4000KB

• 高级程序设计语言都会提供描述__(28)__、__(29)__、控制和数据传输的语言成分，控制成分中有顺序结构、选择结构、__(30)__。

(28)A. 数据　　B. 整型　　C. 数组　　D. 指针

(29)A. 判定　　B. 函数　　C. 运算　　D. 递归

(30)A. 函数　　B. 循环　　C. 递归　　D. 反射

• 在以阶段划分的编译器中，贯穿于编译器工作始终的是__(31)__。

(31)A. 词法分析和语法分析　　B. 语法分析和语义分析

C. 符号表管理和出错处理　　D. 代码优化

• 将一个可执行程序翻译成某种高级程序设计语言源程序的过程称为__(32)__。

(32)A. 编译　　B. 反编译　　C. 汇编　　D. 解释

• 正规式(ab|c)(01|2)表示的正规集合中有__(33)__个元素，__(34)__属于该正规集。

(33)A. 3　　B. 5　　C. 6　　D. 9

(34)A. abc012　　B. a0　　C. c02　　D. c0

• 在函数调用时，引用调用方式下传递的是实参的__(35)__。

(35)A. 左值　　B. 右值　　C. 名称　　D. 类型

• 单链表不具有的特点是__(36)__。

(36)A. 插入、删除运算不需要移动元素

B. 可随机访问链表中的任一元素

C. 不必事先估计存储空间值

D. 所需存储空间量与线性表长度成正比

• 不适合采用栈结构的是__(37)__。

(37)A. 判断一个表达式中的括号是否匹配

B. 判断一个字符串是否是中心对称

C. 按照深度优先的方式后序遍历二叉树

D. 按照层次顺序遍历二叉树

• 设有字符串 S 和 P，串的模式匹配是指__(38)__。

(38)A. 确定 P 在 S 中首次出现的位置　　B. 将 S 和 P 连接起来

C. 将 S 替换为 P　　D. 比较 S 和 P 是否相同

• 以下关于特殊矩阵和稀疏矩阵的叙述中，正确的是__(39)__。

(39)A. 特殊矩阵适合采用双向链表存储，稀疏矩阵适合采用单向链表存储

B. 特殊矩阵的非零元素分布有规律，可以用一维数组进行压缩存储

C. 稀疏矩阵的非零元素分布没有规律，只能用二维数组压缩存储

D. 稀疏矩阵的非零元素分布没有规律，只能用双向链表进行压缩存储

• 已知某二叉树的先序遍历序列为 ABDCEFG、中序遍历序列为 BDACFGE，则该二叉树的层数为__(40)__。

(40)A. 3　　B. 4　　C. 5　　D. 6

• 在一棵非空的二叉排序树中，关键字最大的结点的__(41)__。

(41)A. 左子树一定为空，右子树不一定为空

B. 左子树不一定为空，右子树一定为空

C. 左子树和右子树一定都为空

D. 左子树和右子树一定都不为空

• 为实现快速排序算法，待排序列适合采用__(42)__。

(42)A. 顺序存储　　B. 链式存储　　C. 散列存储　　D. 索引存储

• 若某无向图具有 n 个顶点、e 条边，则其邻接矩阵中值为 0 的元素个数为__(43)__。

(43)A. e　　B. $2e$　　C. $n*n-2e$　　D. n−2e

• Peter Coad 和 Edward Yourdon 将面向对象表示为对象、分类、继承和__(44)__之和。

(44)A. 通过消息的通信　　B. 对象的属性

C. 对象的行为　　D. 对象的抽象

• 在统一建模语言(UML)中，__(45)__展现了一组对象以及它们之间的关系，给出了系统的静态设计视图或静态进程视图，描述了__(46)__中所建立的事物实例的静态快照。

(45)A. 序列图　　B. 状态图　　C. 对象图　　D. 通信图

(46)A. 类图　　B. 组件图　　C. 对象图　　D. 包图

• 继承父类和子类质检共享数据和方法的机制，类的继承支持多态的实现。以下关于类继承的说法中，不正确的是__(47)__。在多态的几种不同的形式中，__(48)__多态是指同一个名字在不同上下文中可代表不同的含义。

(47)A. 一个父类可以有多个子类

B. 父类描述子类的公共属性和方法

C. 一个子类可以继承父类中的属性和方法而不必在子类中定义

D. 子类不可以定义新的属性和方法

(48)A. 参数　　B. 包含　　C. 过载　　D. 强制

• 某教务系统的部分需求包括：教务人员输入课程信息；学生选择课程，经教务人员审核后安排到特定的教室和时间上课；教师根据安排的课程上课，考试后录入课程成绩；学生可以查询本人的成绩；教务人员可以增加、修改、删除和查询课程信息。若用顶层数据流图来建模，则上述需求应包含__(49)__个加工。用模块化方法对系统进行模块划分后，若将对课程信息的增加、修改、删除和查询放到一个模块中，则该模块的内聚类型为__(50)__。

(49)A. 1　　B. 3　　C. 5　　D. 6

(50)A. 逻辑内聚　　B. 信息内聚　　C. 过程内聚　　D. 功能内聚

• 黑盒测试不能发现__(51)__。

(51)A. 不正确或遗漏的功能　　B. 初始化或终止性错误

C. 程序的某条路径存在逻辑错误　　D. 错误的处理结果

• 在软件正式运行后，一般来说，__(52)__错误导致的维护代价最高。

(52)A. 需求　　B. 概要设计　　C. 详细设计　　D. 编码

• 软件测试的原则不包括__(53)__。

(53)A. 测试应在软件项目启动后尽早介入

B. 测试工作应该避免由原开发软件的人或小组承担

C. 测试应该考虑所有的测试用例，确保测试全面性

D. 测试应该严格按照测试计划进行，避免测试的随意性

• 在软件开发过程中，管理者和技术人员的观念是十分重要的。以下叙述中正确的是__(54)__。

(54)A. 如果已经落后于计划，必须增加更多的程序员来赶上进度

B. 在程序真正运行之前，就可以对其设计进行质量评估

C. 有了概要设计就足以开始写程序了，以后可以补充细节

D. 项目需求总是在不断的变化，但这些变化很容易满足，因为软件是灵活的

• 软件开发出现质量问题的主要原因不包括__(55)__。

(55)A. 软件开发人员与用户对应用需求的理解有差异

B. 编程人员与设计人员对设计说明书的理解有差异

C. 软件开发项目的管理有问题

D. 开发软件所用的工具部够先进

• 软件工程每个阶段的各类文档完成后，需要对文档进行复审，这是保证软件产品质量的关键步骤之一。对设计文档进行复审的主要内容不包括__(56)__。

(56)A. 设计文档中对要件的定义是否含糊不清，是否有重复或歧义的定义

B. 设计文档中各项内容是否满足了用户的需求

C. 设计文档是否有利于团队合作实施

D. 对设计文档中所有的要件能否通过测试手段来验证

• 设有公民关系 P(姓名，身份证号，年龄，性别，联系电话，家庭住址)，__(57)__唯一标识关系 P 中的每一个元组，并且应该用__(58)__来进行主键约束。该关系中，__(59)__属于复合属性。

(57)A. 姓名　　B. 身份证号　　C. 联系电话　　D. 家庭住址

(58)A. NULL　　B. NOT NULL　　C. PRIMARY KEY　　D. FOREIGN KEY

(59)A. 姓名　　B. 身份证号　　C. 联系电话　　D. 家庭住址

• 若要将身份证号为“100120189502101111”的人的姓名修改为“刘丽华”，则对应的 SQL 语言为：

```
UPDATE P
__(60)__
WHERE __(61)__ = ' 100120189502101111';
```

(60)A. SET 姓名＝'刘丽华'　　B. Modify 姓名＝'刘丽华'

C. SET 姓名＝刘丽华　　D. Modify 姓名＝刘丽华

(61)A. 刘丽华　　B. '刘丽华'　　C. 身份证号　　D. '身份证号'

• 若要查询家庭住址包含“朝阳区”的人的姓名及联系电话，则对应的 SQL 语句为：

```
SELECT 姓名，电话
FROM P
WHERE 家庭住址 __(62)__ ;
```

(62)A. IN(朝阳区)　　B. like'朝阳区'

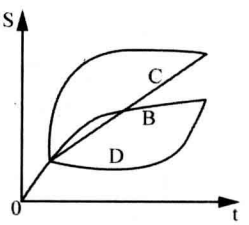
C. IN('朝阳区')　　D. like'%朝阳区%'

• 平面上由条件 X≥0、Y≥0、2X＋Y≤6 和 X＋2Y≤6 所限定的区域，其面积为__(63)__。

(63)A. 2　　B. 3　　C. 4　　D. 6

• 某汽车在匀速行驶一段时间后，司机踩刹车逐渐减速直到停车。为描述其行驶过程，以时间 t 为 X 轴，建立坐标系。下图中，曲线__(64)__大致反映了其刹车过程。

(64)S



• 随着社会信息化程度的迅速提高，我们已经进入了大数据时代。数据量的单位也在不断扩展：B、KB、MB、GB、TB、PB、EB、ZB 等，后者是前者的 1024 倍。因此，1EB＝__(65)__GB。

(65)A. 1K　　B. 1M　　C. 1G　　D. 1T

• ISO/OSI 参考模型的__(66)__使用硬件地址作为服务访问点。

(66)A. 物理层　　B. 数据链路层　　C. 网络层　　D. 传输层

• 以下 IP 地址中，__(67)__可以指定给因特网接口。

(67)A. 10.110.33.224　　B. 40.94.255.10

C. 172.16.17.18　　D. 192.168.22.35

• 在 HTML 中，表格边框的宽度由__(68)__属性指定。

(68)A. width　　B. height　　C. boeder　　D. cellpadaing

• 在地址栏中输入 www.abc.com 浏览器默认的协议是__(69)__。

(69)A. HTTP　　B. DNS　　C. TCP　　D. FTP

• 在 Windows 系统中，通过安装__(70)__组件来创建 FTP 站点。

(70)A. DNS　　B. IIS　　C. POP3　　D. Telnet

• In C language __(71)__ consists of variables and constants connected by operators.

(71)A. an expression　　B. a subroutine

C. a function　　D. a loop

• We cinsider a __(72)__ successful only when an error is discovered.

(72)A. design　　B. program　　C. development　　D. test

• __(73)__ Of database refers to the protection of data against unauthorized disclosure, alteration, or destruction.

(73)A. Security　　B. Access　　C. Backup　　D. Creation

• One of the major features in C++ is __(74)__ handling, which is a better way of handling errors.

(74)A. data　　B. pointer　　C. test　　D. exception

• __(75)__ is a method or procedure for carrying out a task.

(75)A. Thought　　B. Ideality　　C. Algorithm　　D. Creation

下午试卷

(考试时间 150 分钟，满分 75 分)

本试卷的试题中共有 5 题，每题 15 分，满分 75 分。

试题一

阅读以下说明和流程图，填补流程图中的空缺(1)～(5)，将解答填入答题纸的对应栏内。

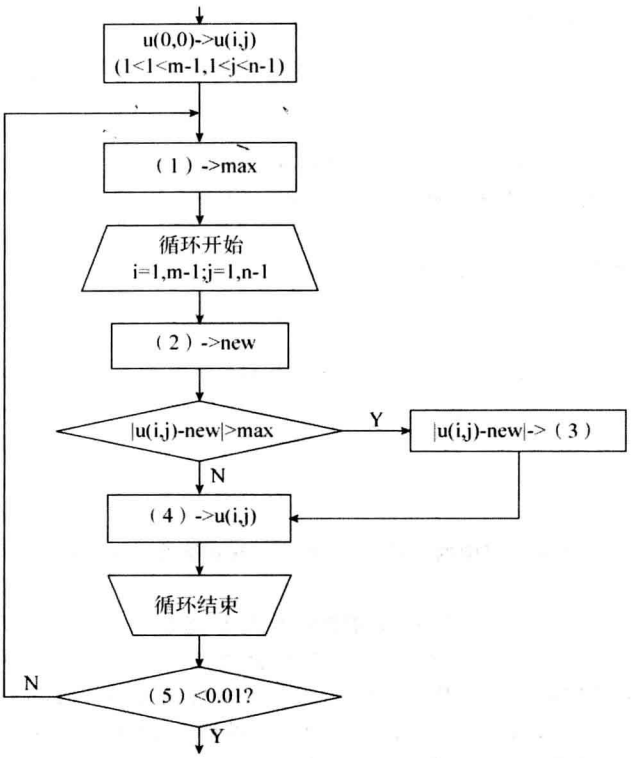
【说明】

平面上一个封闭区域内稳定的温度函数式一个调和函数，如果区域边界上各点的温度是已知的(非常数)，那么就可以用数值方法近似地计算出区域内各点的温度(非负数)。

假设封闭区域是矩形，可将整个矩形用许多横竖线切分成比较细小的网格，并以最简单的方式建立坐标系统，从而可以将问题描述为：已知调和函数 u(i,j)在矩形{0≤i≤m;0≤j≤n}四边上的值，求函数 u 在矩形内部各个网格点{i=1,…,m−1;j=1…,n−1}上的近似值。

根据调和函数的特点可以推导出近似算式：该矩形内任一网格点上的函数值等于其上下左右四个相邻网格点上函数值的算术平均值。这样，我们就可以用迭代法来进行数值计算了。首先将该矩形内各网格点上的函数值设置为一个常数，例如 u(0,0)；然后通过该迭代式计算矩形内个网格点上的新值。这样反复进行迭代计算，若某次迭代后所有的新值与原值之差别都小于预定的要求(例如 0.01)，则结束求解过程。

【流程图】



试题二

阅读以下说明和C函数，填充函数中的空缺，将解答填入答题纸的对应栏内。

【说明】

函数GetDateId(DATE date)的功能是计算并返回指定合法日期date是其所在年份的第几天。例如，date表示2008年1月25日时，函数的返回值为25，date表示2008年3月3日时，函数返回值为63。

函数Kday_Date(int theyear, int k)的功能是计算并返回指定合法年份theyear(theyear≥1900)的第k天(1≤k≤365)所对应的日期。例如，2008年的第60天是2008年2月29日，2009年的第60天是2009年3月1日。

函数isLeapYear(int y)的功能是判断y代表的年份是否为闰年，是则返回1，否则返回0。

DATE类型定义如下：

```
typedef struct{
      int year, month, day;
}DATE;
```

【C函数1】

```
int GetDateId(DATE date)
{
    Const int days_month[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
int i, date_id = date.day;
for(i = 0;i<__(1)__;i++)
    date_id += days_month[i];
if(__(2)__ &&isLeapYear(date.year)) date_id++;
return date_id;
}
```

【C 函数 2】

```
__(3)__ Kday_Date(int theyear, int k)
{
    int i;
    DATE date;
int days_month[13] = {0,31,28,31,30,31,30,31,31,30,31,30,31};
    assert(k>=1 && k<=365 && theyear>=1900); /*不满足断言时程序终止*/
    date.year = __(4)__;
    if(isLeapYear(date.year)) days_month[2]++;
    for(i=1;;){
        k = k - days_month[i++];
        if(k<=0){date.day = k + __(5)__; date.month = i-1; break;}
    }
    return date;
}
```

试题三

阅读以下说明和 C 程序，填充程序中的空缺，将解答填入答题纸的对应栏内。

【说明】

埃拉托斯特尼筛法求不超过自然数 N 的所有素数的做法是：先把 N 个自然数按次序排列起来，1 不是素数，也不是合数，要划去；2 是素数，取出 2(输出)，然后将 2 的倍数都划去；剩下的数中最小者为 3，3 是素数，取出 3(输出)，再把 3 的倍数都划去；剩下的数中最小者为 5，5 是素数(输出)，再把 5 的倍数都划去。这样一直做下去，就会把不超过 N 的全部合数都筛掉，每次从序列中取出的最小数构成的序列就是不超过 N 的全部质数。

下面的程序实现埃拉托斯特尼筛法求素数，其中，数组元素 sieve[i](i>0)的下标 i 对应自然数 i，sieve[i]的值为 1/0 分别表示 i 在/不在序列中，也就是将 i 划去(去掉)时，就将 sieve[i]设置为 0。

【C 程序】

```
#include<stdio.h>
#define N 10000
int main()
{
   char sieve[N+1] = {0};
   int i=0,k;
   /*初始时 2～N 都放入 sieve 数组*/
   for(i=2; __(1)__ ;i++)
   sieve[i] = 1;
   for(k=2; ;){
   /*找出剩下的数中最小者并用 K 表示*/
   for( ; k<N+1 && sieve[k] == 0; __(2)__ );

   if( __(3)__ ) break;
   print("%d\t", k); /*输出素数*/

   /*从 sieve 中去掉 k 及其倍数*/
   for(i=k; i<N+1; i= __(4)__ )
   __(5)__;
   }
```

```
    return 0;
} /* end of main */
```

试题四

阅读以下说明和C程序，填充函数中的空缺，将解答填入答题纸的对应栏内。

【说明】

N 个游戏者围成一圈，从 1～N 顺序编号，游戏方式如下：从第一个人开始报数（从 1 到 3 报数），凡报到 3 的人退出圈子，直到剩余一个游戏者为止，该游戏者即为获胜者。

下面的函数 playing(Linklist head)模拟上述游戏过程并返回获胜者的编号。其中，N 个人围成的圈用一个包含 N 个结点的单循环链表来表示，如图 4-1 所示，游戏者的编号放在结点的数据域中。

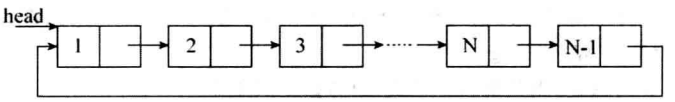


图 4-1

在函数中，以删除结点来模拟游戏者退出圈子的处理。整型变量 c（初值为 1）用于计数，指针变量 p 的初始值为 head，如图 4-1 所示。游戏时，从 p 所指向的结点开始计数，p 沿链表中的指针方向遍历结点，c 的值随 p 的移动相应地递增。当 c 计数到 2 时，就删除 p 所指结点的下一个结点（因下一个结点就表示报数到 3 的游戏者），如图 4－2 所示，然后将 c 设置为 0 后继续游戏过程。

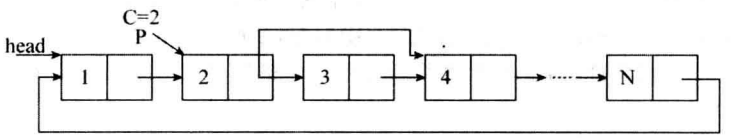


图 4-2

结点类型定义如下：

```
typedef struct node{
    int code;        /* 游戏者的编号 */
    struct node *next;
}NODE, *LinkList;
```

【C 函数】

```
int playing(LinkList head,int n)
{ /* head 指向含有 n 个结点的循环单链表的第一个结点（即编号为 1 的游戏者）*/
LinkList p = head, q;
int thewinner, c = 1;

while(n> __(1)__ ){
    if(c = = 2){     /* 当 c 等于 2 时，p 所指向结点的后继即为将被删除的结点 */
q = p->next;
p->next = __(2)__ ;
printf("%d\t", q->code); /* 输出退出圈子的游戏者编号 */
free(q);
c = __(3)__ ;
n--;
} /* if */
p = __(4)__ ;
c++;
```

```
    } /* while */
    theWinner = __(5)__;
    free(p);

return theWinner; /* 返回最后一个游戏者(即获胜者)的编号 */
}
```

试题五

阅读下列说明和C++代码，填充代码中的空缺，将解答填入答题纸的对应栏内。

【说明】

某学校在学生毕业时要求对其成绩进行综合评定，学生的综合成绩(GPA)由其课程加权平均成绩(Wg)与附加分(Ag)构成，即 GPA = Wg + Ag。

设一个学生共修了 n 门课程，则其加权平均成绩(Wg)定义如下：

其中，$grade_i$、C_i 分别表示该学生第 i 门课程的百分制成绩及学分。

学生可以通过参加社会活动或学科竞赛获得附加分(Ag)。学生参加社会活动所得的活动分(Apoints)是直接给出的，而竞赛分(Awards)则由下式计算(一个学生最多可参加 m 项学科竞赛)：

其中，l_i 和 s_i 分别表示学生所参加学科竞赛的级别和成绩。

对于社会活动和学科竞赛都不参加的学生，其附加分按活动分为 0 计算。

下面的程序实现计算学生综合成绩的功能，每个学生的基本信息由抽象类 Student 描述，包括学号(stuNo)、姓名(name)、课程成绩学分(grades)和综合成绩(GPA)等，参加社会活动的学生由类 ActStudent 描述，其活动分由 Apoints 表示，参加学科竞赛的学生由类 CmpStudent 描述，其各项竞赛的成绩信息由 awards 表示。

【C++代码】

```
#include<string>
#include<iostream>
using namespace std;
const int n = 5;                    /* 课程数 */
const int m = 2;                    /* 竞赛项目数 */

class Student{
protected:
    int stuNo; string name;
    double GPA;                     /* 综合成绩 */
    int(*grades)[2];                /* 各门课程成绩和学分 */
public:
    Student(const int stuNo, const string&name, int grades[][2]){
        this->stuNo = stuNo; this->name = name; this->grades = grades;
}
Virtual~Student(){}
int getstuNo() {/* 实现略 */}
string getName() {/* 实现略 */}
  __(1)__;
double computeWg(){
    int totalGrades = 0, totalCredits = 0;
    for(int i = 0; i<N; i++){
        totalGrades += grades[i][0] * grades[i][1]; totalGredits += grades[i][1];
    }
```

```
    return GPA = (double)totalGrades/totalCredits;
  }
};
class ActStudent: public Student{
int Apoints;
public:
  ActStudent(const int stuNo, const string &name, int gs[][2], int Apoints)
: __(2)__ {
    this->Apoints = Apoints;
  }
  double getGPA(){ return GPA = __(3)__;}
};

    class CmpStudent: public Student{
    private:
        int(*awards)[2];
    public:
        CmpStudent(const int stuNo,const string &name, int gs[][2], int awards[][2])
        : __(4)__ {    this->award = award; }
        double getGPA(){
            int Awards = 0;
            for(int i = 0; i<M; i++){
                Awards += awards[i][0] * awards[i][1];
            }
            Return GPA = __(5)__;
        }
    };
int main()
{    //以计算3个学生的综合成绩为例进行测试
     int g1[][2] = {{80,3},{90,2},{95,3},{85,4},{86,3}},
       g2[][2] = {{60,3},{60,2},{60,3},{60,4},{65,3}},
       g3[][2] = {{80,3},{90,2},{70,3},{65,4},{75,3}};          //课程成绩
     int c3[][2] = {{2,3},{3,3}};                               //竞赛成绩
     Student *student[3] = {
         new ActStudent(101,"John",g1,3),                       //3为活动分
         new ActStudent(102,"Zhang",g2,0),
         new ActStudent(103,"Li",g3,c3),
     };

     //输出每个学生的综合成绩
     for(int i = 0;i<3;i++)
         cout<< __(6)__ <<endl;
     delete *student;
     return 0;
}
```