

全国计算机技术与软件专业技术资格（水平）考试

2008 年上半年 程序员 上午试卷

（考试时间 9：00～11：30 共 150 分钟）

请按下述要求正确填写答题卡

1. 在答题卡的指定位置上正确写入你的姓名和准考证号，并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。
2. 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。
3. 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。
4. 解答前务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项，如需修改，请用橡皮擦干净，否则会导致不能正确评分。

例题

● 2008 年上半年全国计算机技术与软件专业技术资格（水平）考试日期是
____（88）____月____（89）____日。

- | | | | | |
|------|-------|-------|-------|-------|
| （88） | A. 4 | B. 5 | C. 6 | D. 7 |
| （89） | A. 21 | B. 22 | C. 23 | D. 24 |

因为考试日期是“5 月 24 日”，故（88）选 B，（89）选 D，应在答题卡序号 88 下对 B 填涂，在序号 89 下对 D 填涂（参看答题卡）。

● 在 Word 编辑状态下，可以把一段文字的格式快速应用到另一段文字上，以提高编排文章的效率，其方法是：单击常用工具栏上的“(1)”按钮；然后移动鼠标到待格式化的文本开始处，(2)。

- (1) A.  B.  C.  D. 

- (2) A. 按<Alt>键的同时单击鼠标左键即可
B. 按<Alt>键的同时单击鼠标右键即可
C. 按住鼠标左键拖动鼠标，到达待格式化的文本末尾时松开鼠标键
D. 按住鼠标右键拖动鼠标，到达待格式化的文本末尾时松开鼠标键

● Excel 单列表格(3)可以根据“分隔符号”分列成多列表格。如果选中某单元格并输入 2000，回车后此单元格的显示内容为¥2000，那么应将此单元格的格式设置成(4)。

- (3) A.

	A
1	城市 第一季度第二季度第三季度第四季度
2	北京¥1500.00¥1800.00¥2000.00¥1900.00
3	上海¥1800.00¥2000.00¥2600.00¥2100.00
4	西安¥1600.00¥1800.00¥1300.00¥1600.00

- B.

	A
1	城市, 第一季度, 第二季度, 第三季度, 第四季度
2	北京, ¥1500.00, ¥1800.00, ¥2000.00, ¥1900.00
3	上海, ¥1800.00, ¥2000.00, ¥2600.00, ¥2100.00
4	西安, ¥1600.00, ¥1800.00, ¥1300.00, ¥1600.00

- C.

	A
1	城市 ; 第一季度; 第二季度; 第三季度; 第四季度
2	北京, ¥1500.00, ¥1800.00, ¥2000.00, ¥1900.00
3	上海, ¥1800.00, ¥2000.00, ¥2600.00, ¥2100.00
4	西安, ¥1600.00, ¥1800.00, ¥1300.00, ¥1600.00

- D.

	A
1	城市, 第一季度, 第二季度, 第三季度, 第四季度
2	北京; ¥1500.00; ¥1800.00; ¥2000.00; ¥1900.00
3	上海; ¥1800.00; ¥2000.00; ¥2600.00; ¥2100.00
4	西安; ¥1600.00; ¥1800.00; ¥1300.00; ¥1600.00

- (4) A. 数值 B. 人民币 C. 货币 D. 科学记数

● 因特网最高层域名分为机构性域名和地理性域名两大类，目前主要有 14 种机构性域名，其中“(5)”是教育机构域名。

- (5) A. COM B. EDU C. GOV D. ORG

● 将十六进制数 9B 转换为八进制数为(6)。

- (6) A. 233 B. 433 C. 463 D. 531

● 冯·诺依曼体系结构的计算机有两个主要组成部件：内存和处理器。其中，内存用

于存储指令和数据。在大多数高级语言程序中，(7)是内存单元的抽象。微处理器中的 ALU 可执行算术运算和(8)操作。

- (7) A. 变量 B. 关键字 C. 语句 D. 数据类型
(8) A. 浮点 B. 定点 C. 逻辑 D. 控制

● 已知某字符的编码为“0100101”，若最高位增加一个偶校验位，则其编码变为(9)。

- (9) A. 10100101 B. 11001010 C. 01000110 D. 01010101

● 对于一个基于网络的应用系统，在客户端持续地向服务端提交作业请求的过程中，若作业响应时间越短，则服务端(10)。

- (10) A. 占用内存越大 B. 越可靠
C. 吞吐量越大 D. 抗病毒能力越强

● 在软件开发中使用函数库可(11)。

- (11) A. 提高软件的执行速度 B. 降低系统负载
C. 提高软件各部分之间的耦合度 D. 提高软件的可重用性

● 在显存中，表示黑白图像的像素点最少需(12)位。

- (12) A. 1 B. 2 C. 8 D. 16

● 图像文件格式可以分为静态格式和动态格式，(13)属于动态图像文件格式。

- (13) A. BMP B. PNG C. WAV D. AVI

● (14)图像通过使用彩色查找表来获得图像颜色。

- (14) A. 真彩色 B. 伪彩色 C. 黑白 D. 矢量

● 计算机(15)感染可执行文件。一旦直接或间接执行了被该病毒感染的程序，该病毒会按照编制者的意图对系统进行破坏。

- (15) A. 文件型病毒 B. 宏病毒
C. 目录病毒 D. 引导型病毒

● 为防止攻击者通过 Windows XP 系统中正在运行的 Telnet 服务登录到用户的计算机，可以对 Windows XP 中 Telnet 服务进行如下设置：(16)。

- (16) A. 设置启动类型为自动 B. 设置启动类型为自动，之后重启计算机
C. 设置启动类型为手动 D. 设置启动类型为手动，之后重启计算机

● 依据我国著作权法的规定，(17)不可转让，不可被替代，不受时效的约束。

- (17) A. 翻译权 B. 署名权 C. 修改权 D. 复制权

● 甲委托乙开发一应用软件，若双方没有订立任何书面合同，那么该软件著作权应由 (18) 。

- (18) A. 乙一人享有 B. 甲、乙共同享有
C. 甲一人享有 D. 甲、乙均不享有

● 设机器码的长度为 8， x 为带符号纯小数， y 为带符号纯整数， $[X]_{\text{原}} = 11111111$ ， $[Y]_{\text{补}} = 11111111$ ，则 x 的十进制真值为 (19)， y 的十进制真值为 (20)。

- (19) A. 1/128 B. -1/128 C. -127/128 D. 127/128
(20) A. -1 B. 127 C. -127 D. 1

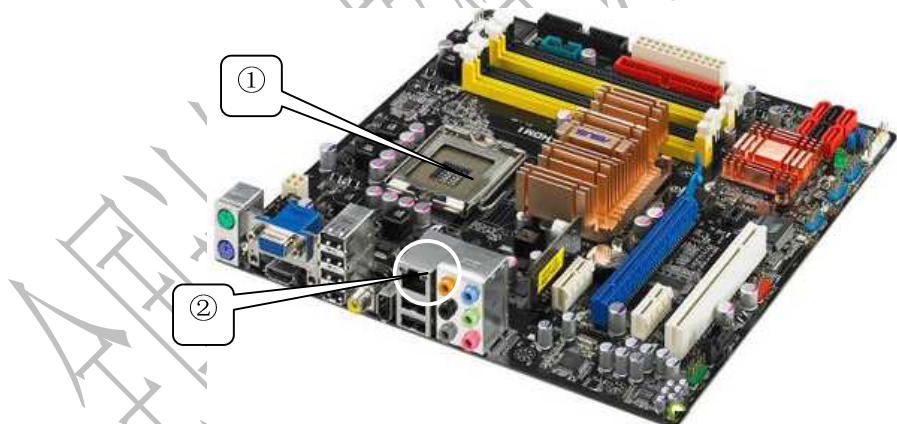
● 使用 (21) 技术，计算机的微处理器可以在完成一条指令前就开始执行下一条指令。

- (21) A. 迭代 B. 流水线 C. 面向对象 D. 中间件

● 微机启动时使用的有关计算机硬件配置的重要参数保存在 (22) 中。

- (22) A. Cache B. CMOS C. RAM D. CD-ROM

● 下图所示的微机主板的①处是 (23) 插槽，②处是 (24) 接口。



- (23) A. CPU B. PCI C. RAM D. IDE
(24) A. COM B. RJ-45 C. USB D. PS/2

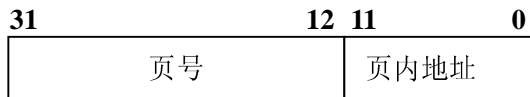
● 操作系统的任务是 (25) 。

- (25) A. 把源程序转换为目标代码 B. 管理计算机系统上的软、硬件资源
C. 负责存取数据库中的各种数据 D. 负责文字格式编排和数据计算

● 若进程 P1 正在运行，操作系统强行撤下 P1 所占用的 CPU，让具有更高优先级的进程 P2 运行，此时 P1 进程处于 (26) 状态。

- (26) A. 就绪 B. 等待 C. 结束 D. 善后处理

● 某页式存储管理系统中的地址结构如下图所示，则 (27)。



- (27) A. 页的大小为 1K, 最多有 8M 页
B. 页的大小为 2K, 最多有 4M 页
C. 页的大小为 4K, 最多有 1M 页
D. 页的大小为 8K, 最多有 2M 页

● 关于汇编语言，以下叙述中正确的是 (28)。

- (28) A. 汇编语言源程序可以直接在计算机上运行
B. 将汇编语言源程序转换成目标程序的软件称为解释程序
C. 在汇编语言程序中，不能定义符号常量
D. 将汇编语言源程序翻译成机器语言程序的软件称为汇编程序

● 对高级语言源程序进行编译时，可发现源程序中的 (29) 错误。

- (29) A. 堆栈溢出 B. 变量未定义 C. 指针异常 D. 数组元素下标越界

● 计算机程序=算法+数据结构+程序设计方法+语言工具和环境，其中关于程序设计的叙述，正确的是 (30)。

- (30) A. 程序设计语言与程序设计方法是一一对应的关系
B. 面向对象的程序设计语言只能支持面向对象的程序设计方法
C. 面向对象的程序设计技术与结构化程序设计技术是相互排斥的
D. 过程式程序设计是一种传统的程序设计方法

● 计算机能直接识别和执行机器语言程序，该语言的基本元素是 (31)。

- (31) A. 汇编代码 B. 0 和 1 C. 扩展 ASCII 码 D. BCD 码

● 若二维数组 P[1..5, 0..8]的首地址为 base，数组元素按行存储，且每个元素占用 1 个存储单元，则元素 P[3, 3]在该数组空间的地址为 (32)。

- (32) A. base+13 B. base+16 C. base+18 D. base+21

● 设初始栈为空，s 表示入栈操作，x 表示出栈操作，则 (33) 是合法的操作序列。

- (33) A. sxxssssxxx B. xxssssxxs C. sxxssssxx D. xssssssxx

● 在下面的程序代码中，a 是全局变量。若以引用调用（call by reference）方式调用函数 f，则输出结果为 (34)；若以值调用（call by value）方式调用函数 f，输出结果为 (35)。

Main program

```
a:=5;;
b:=f(a);
write(a+b);
```

function f(x)

```
a:=a-1;
x:=a+x;
return a*x;
```

(34) A. 10

B. 40

C. 36

D. 72

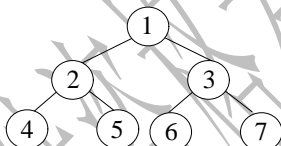
(35) A. 10

B. 40

C. 36

D. 72

● 满二叉树的特点是每层上的结点数都达到最大值，因此对于高度为 h ($h > 1$) 的满二叉树，其结点总数为 (36)。对非空满二叉树，由根结点开始，按照先根后子树、先左子树后右子树的次序，从 1、2、3、…依次编号，则对于树中编号为 i 的非叶子结点，其右子树的编号为 (37)（高度为 3 的满二叉树如下图所示）。



(36) A. 2^h

B. 2^{h-1}

C. $2^h - 1$

D. $2^{h-1} + 1$

(37) A. $2i$

B. $2i-1$

C. $2i+1$

D. $2i+2$

● 在数据结构中，结点（数据元素）及结点间的相互关系组成数据的逻辑结构。按逻辑结构的不同，数据结构通常可分为 (38) 两类。

(38) A. 线性结构和非线性结构

B. 紧凑结构和稀疏结构

C. 动态结构和静态结构

D. 内部结构和外部结构

● 采用哈希（或散列）技术构造查找表时，需要考虑冲突（碰撞）的处理，冲突是指 (39)。

(39) A. 关键字相同的记录被映射到不同的哈希地址

B. 关键字依次被映射到编号连续的哈希地址

C. 关键字不同的记录被映射到同一个哈希地址

D. 关键字的数目超过哈希地址的数目

● 数据结构中的树最适合用来表示 (40) 的情况。

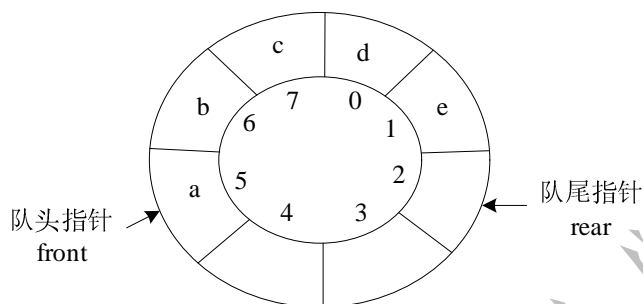
(40) A. 数据元素有序

B. 数据元素之间具有多对多关系

C. 数据元素无序

D. 数据元素之间具有一对多关系

● 某循环队列的容量为 M ，队头指针指向队头元素，队尾指针指向队尾元素之后，如下图所示 ($M=8$)，则队列中的元素数目为 (41) (MOD 表示整除取余运算)。



- (41) A. $\text{rear} - \text{front}$ B. $\text{front} - \text{rear}$
C. $(\text{rear} - \text{front} + M) \text{MOD } M$ D. $(\text{front} - \text{rear} + M) \text{MOD } M$

● 二叉排序树或者是一棵空树，或者是具有如下性质的二叉树：若其左子树非空，则左子树上所有结点的值均小于根结点的值；若其右子树非空，则右子树上所有结点的值均大于根结点的值；其左、右子树本身就是两棵二叉排序树。根据该定义，对一棵非空的二叉排序树进行 (42) 遍历，可得到一个结点元素的递增序列。

- (42) A. 先序 (根、左、右)
B. 中序 (左、根、右)
C. 后序 (左、右、根)
D. 层序 (从树根开始，按层次)

● 对于 n 个元素的关键字序列 $\{k_1, k_2, \dots, k_n\}$ ，若将其按次序对应到一棵具有 n 个结点的完全二叉树上，使得任意结点都不大于其孩子结点 (若存在孩子结点)，则称其为小顶堆。根据以上定义，(43) 是小顶堆。

- (43) A. B. C. D.

● 下列关于结构化设计与面向对象设计的叙述中，不正确的是 (44)。

- (44) A. 结构化设计以过程设计为核心
B. 面向对象设计以对象设计为核心
C. 结构化设计比面向对象设计更易支持数据结构的修改
D. 面向对象设计比结构化设计更易支持系统的维护

● 下面关于类、对象和实例的叙述中，错误的是(45)。

- (45) A. 类是创建对象的模板 B. 对象是类的实例
C. 类是对象的实例 D. 类是一组具有共同特征的对象集合

● (46)不是面向对象程序设计语言。

- (46) A. XML B. Java C. C# D. Simula

● 面向对象中的(47)机制是对现实世界中遗传现象的模拟，通过该机制，基类的属性和方法被遗传给派生类。

- (47) A. 封装 B. 多态 C. 继承 D. 变异

● UML 是一种(48)。

- (48) A. 面向对象的程序设计语言 B. 面向过程的程序设计语言
C. 软件系统开发方法 D. 软件系统建模语言

● 软件设计活动中的(49)设计指定各个组件之间的通信方式以及各组件之间如何相互作用。

- (49) A. 数据 B. 接口 C. 结构 D. 组件

● 原型化方法是动态确定软件需求的方法之一，该方法适应于(50)的系统。

- (50) A. 需求不确定性高 B. 需求确定 C. 结构简单 D. 可移植性好

● 瀑布模型是传统的软件开发过程模型，它强调各阶段的严格性，其主要缺点是(51)。

- (51) A. 需要软件人员和用户进行沟通 B. 需要付出较高的维护成本
C. 开发的软件不易于移植 D. 不适应需求不确定的软件开发

● 将软件从一种计算机环境转换到另一种环境运行的难易程度是指软件的(52)。
在规定的条件下和规定的时间间隔内，按设计要求，软件成功运行的特性称为(53)。

- (52) A. 可靠性 B. 可移植性 C. 可使用性 D. 灵活性
(53) A. 可靠性 B. 可移植性 C. 可重用性 D. 正确性

● 软件测试是保证软件质量的关键活动，实施软件测试的可行目标是(54)。

- (54) A. 证明软件不存在错误和缺陷
B. 验证程序的数据表示方式是否正确
C. 找出软件中的所有错误和缺陷
D. 尽量多地发现软件中的错误和缺陷

● 软件测试通常采用黑盒测试和白盒测试。其中黑盒测试根据软件的_(55)_设计测试用例，白盒测试根据软件的_(56)_设计测试用例。

- (55) A. 功能规格说明 B. 需求说明 C. 内部结构和逻辑 D. 数据流图
(56) A. 功能规格说明 B. 需求说明 C. 内部结构和逻辑 D. 数据流图

● 数据库是在计算机系统中按照一定的数据模型组织、存储和应用的_(57)_的集合。提供数据库各种操作的软件系统叫做_(58)_。

- (57) A. 命令 B. 程序 C. 数据 D. 文件
(58) A. 数据库系统 B. 文件系统 C. 操作系统 D. 数据库管理系统

● 对于下图所示的 Students 和 SC 关系，属性 Sno 表示学号，Sname 表示姓名，Sex 表示性别，SD 表示专业，Age 表示年龄，Cno 表示课程号，Grade 表示成绩。

Sno	Sname	Sex	SD	Age	Sno	Cno	Grade
310001	张 静	女	计算机	18	310002	1	93
310002	陈继军	男	计算机	19	310002	2	84
410003	李丽莉	女	机 械	18	310002	3	84
410004	刘华东	男	机 械	19	410004	2	83
110041	赵庆民	男	通 信	20	410004	3	93
110042	范建华	男	通 信	20	110042	1	84
					110042	2	82

Students 关系

SC 关系

a. 当用户查询“选了课程 2 的学生数”时，其结果应为_(59)_；当 Students 和 SC 关系进行自然连接时，其结果集的属性列数应为_(60)_。

- (59) A. 3 B. 4 C. 6 D. 7
(60) A. 6 B. 7 C. 8 D. 9

b. 查询“至少选修了两门课程的每个学生的平均成绩”的 SQL 语句如下：

```
SELECT Sno, (61)
FROM SC
GROUP BY Sno
(62) >1;
```

- (61) A. Grade B. SUM (Grade) / 7
C. AVG (Grade) D. SUM (Grade) / COUNT (Cno)
(62) A. WHERE COUNT (Sno)
B. WHERE COUNT (DISTINCT (Sno))
C. HAVING COUNT (Sno)
D. HAVING COUNT (DISTINCT (Sno))

● 软件开发工期以及所需的开发人数依赖于软件规模以及开发方式。有人对某种开发方式下软件的规模（以千行源代码为单位）、开发工期（月数）以及平均所需的开发人

数做了统计，如下表所示：

软件规模 (KDSI)	开发工期 (月)	平均所需的开发人数
2	4	1.1
8	8	2.7
32	14	6.5
128	24	16

对于规模为 12.8 千行源代码 (KDSI) 的软件，利用上表以及线性插值方法，可以估算出，这种开发方式所需的开发工期 (月数) 及开发人数为 (63)。

- (63) A. 9.0, 3.0 B. 9.2, 3.5 C. 9.5, 3.8 D. 9.7, 4.0

● 为了用二分法求函数 $f(x)=x^3-2x^2-0.1$ 的根 (方程 $f(x)=0$ 的解)，可以选择初始区间 (64)。也就是说，通过对该区间逐次分半可以逐步求出该函数的一个根的近似值。

- (64) A. $[-2, -1]$ B. $[-1, 1]$ C. $[1, 2]$ D. $[2, 3]$

● 设任意多面体的顶点数为 V ，边数为 E ，面数为 F 。请根据实例判断并选出正确反映这三者之间关系的公式 (65)。

- (65) A. $V+E=F+2$ B. $V+F=E+2$ C. $E*F=V+10$ D. $E+F=V+10$

● 在 TCP/IP 传输层的段 (segment) 中，采用 (66) 来区分不同的应用进程。

- (66) A. 端口号 B. IP 地址 C. 协议类型 D. MAC 地址

● 在检测网络故障时使用的 ping 命令是基于 (67) 协议实现的。

- (67) A. SNMP (简单网络管理协议)
B. FTP (文件传输协议)
C. IGMP (互联网组管理协议)
D. ICMP (互联网控制管理协议)

● 在 Outlook 中，通常借助 (68) 来传送一个文件。

- (68) A. 邮件正文 B. Telnet C. WWW D. 附件功能

● 在 Windows Server2003 操作系统中可以通过安装 (69) 组件创建 FTP 站点。

- (69) A. IIS B. IE C. POP3 D. DNS

● 某主机的 IP 地址为 10.100.100.01/20，其子网掩码是 (70)。

- (70) A. 255.255.248.0 B. 255.255.252.0
C. 255.255.240.0 D. 255.255.255.0

● The line of computing jobs waiting to be run on a computer system might be a (71). The jobs are serviced in the order of their arrival, that is, the first in is the first out.

- (71) A. queue B. stack C. array D. record

● The (72) is a collection of modules serving as interface between hardware and software to provide a software platform.

- (72) A. computer B. processor
C. application software D. operating system

● C++ fully supports (73) programming.

- (73) A. visual B. object-oriented
C. logic D. natural language

● The Internet is an open system, and most information is available to everyone. It is very important to add (74) to existing system for protecting the secret information.

- (74) A. Web pages B. modem C. security D. network

● A (75) system placed between the company network and the outside world may limit outside access to the internal network.

- (75) A. firewall B. modem C. gateway D. key

全国计算机技术与软件专业技术资格（水平）考试

2008 年上半年 程序员 下午试卷

（考试时间 14:00~16:30 共 150 分钟）

请按下述要求正确填写答题纸

1. 在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。
2. 在答题纸的指定位置填写准考证号、出生年月日和姓名。
3. 答题纸上除填写上述内容外只能写解答。
4. 本试卷共 7 道题，试题一至试题四是必答题，试题五至试题七选答 1 道。
每题 15 分，满分 75 分。

试题号	一~四	五~七
选择方法	必答题	选答 1 题

5. 解答时字迹务必清楚，字迹不清时，将不评分。
6. 仿照下面例题，将解答写在答题纸的对应栏内。

例题

2008 年上半年全国计算机技术与软件专业技术资格（水平）考试日期是(1)月(2)日。

因为正确的解答是“5 月 24 日”，故在答题纸的对应栏内写上“5”和“24”（参看下表）。

例题	解答栏
(1)	5
(2)	24

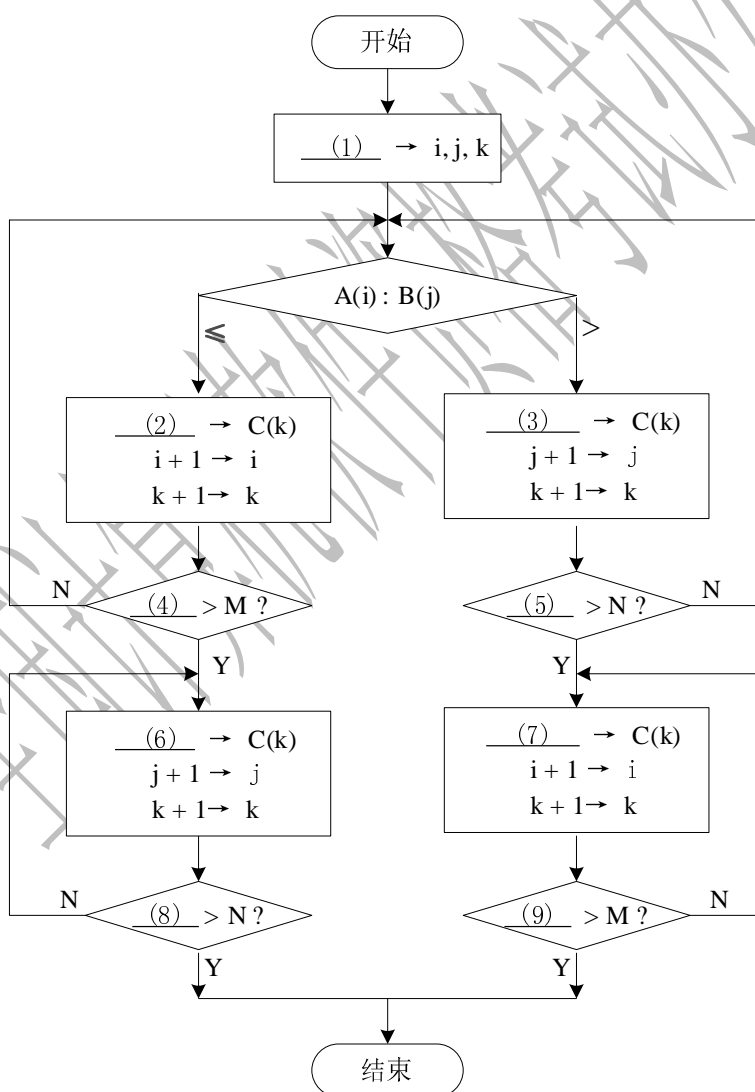
试题一（共 15 分）

阅读以下说明和流程图，填补流程图中的空缺（1）～（9），将解答填入答题纸的对应栏内。

[说明]

假设数组 A 中的各元素 $A(1), A(2), \dots, A(M)$ 已经按从小到大排序 ($M \geq 1$)；数组 B 中的各元素 $B(1), B(2), \dots, B(N)$ 也已经按从小到大排序 ($N \geq 1$)。执行下面的流程图后，可以将数组 A 与数组 B 中所有的元素全都存入数组 C 中，且按从小到大排序（注意：序列中相同的数全部保留并不计排列顺序）。例如，设数组 A 中有元素：2, 5, 6, 7, 9；数组 B 中有元素：2, 3, 4, 7；则数组 C 中将有元素：2, 2, 3, 4, 5, 6, 7, 7, 9。

[流程图]



试题二（共 15 分）

阅读以下说明和 C 程序，将应填入__(n)__处的字句写在答题纸的对应栏内。

[说明]

下面的程序按照以下规则输出给定名词的复数形式：

- 若名词以“y”结尾，则删除 y 并添加“ies”；
- 若名词以“s”、“ch”或“sh”结尾，则添加“es”；
- 其他所有情况，直接添加“s”。

[C 程序]

```
#include <stdio.h>
#include <string.h>
char *plural(char *word)
{
    int n;
    char *pstr;
    n = strlen(word);          /*求给定单词的长度*/
    pstr = (char *)malloc(n+3); /*申请给定单词的复数形式存储空间*/
    if (!pstr || n < 2)
        return NULL;
    strcpy(pstr, word);        /*复制给定单词*/
    if (__(1)__)
    {
        pstr[n-1] = 'i'; pstr[n] = 'e'; pstr[n+1] = 's'; __(2)__;
    }
    else
        if (pstr[n-1]=='s' || pstr[n-1]=='h' && (__(3)__))
        {
            pstr[n] = 'e'; pstr[n+1] = 's'; pstr[n+2] = '\0';
        }
        else
            { pstr[n] = 's'; pstr[n+1] = '\0'; }
    __(4)__;
}

main( )
{
    int i; char *ps;
    char wc[9][10] =
        {"chair", "dairy", "boss", "circus", "fly", "dog", "church", "clue", "dish"};
    for(i = 0; i < 9; i++) {
        ps = __(5)__;
        printf("%s:  %s\n", wc[i], ps); /*输出单词及其复数形式*/
        free(ps);                       /*释放空间*/
    }
    system("pause");
}
```

试题三（共 15 分）

阅读以下说明和 C 程序，将应填入__(n)__处的字句写在答题纸的对应栏内。

[说明]

下面的程序用 Dole Rob 算法生成 N 阶（N 为奇数）魔方阵（各行、列、对角线数字之和相等）。该算法的过程为：从 1 开始，按如下方法依次插入各自然数，直到 N^2 为止：

- 在第一行的正中插入 1；
- 新位置应当处于最近插入位置的右上方，若该位置已超出方阵的上边界，则新位置取应选列的最下一个位置；若超出右边界，则新位置取应选行的最左一个位置；
- 若最近插入的元素是 N 的整数倍，则选同列的下一行位置为新位置。

例如，3 阶魔方阵如下所示：

8	1	6
3	5	7
4	9	2

[C 程序]

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 50
main()
{
    int row, col, n, value;
    int a[SIZE+1][SIZE+1]; /*不使用下标为 0 的元素*/
    printf("请输入要输出魔方阵的阶数 n(奇数, <%d):n=", SIZE);
    scanf("%d",&n);
    if (!(n % 2) || n < 1 || __(1)__) {
        printf("输入数据有误!\n"); exit(0);
    }
    row = 1; col = (n+1)/2; value = 1;
    while(value <= __(2)__) {
        a[row][col] = value;
        /*计算下一位置*/
        if(value%n != 0) {
            row--; __(3)__;
            if(row < 1) row = n;
            if(col > n) __(4)__;
        }
        else row++;
        value = __(5)__;
    }
    printf("\n%d 阶魔方阵如下所示:\n\n",n);
    for(row = 1; row <= n; row++) {
        for(col = 1; col <= n; col++)
            printf("%5d",a[row][col]);
        printf("\n");
    }
}
```

试题四（共 15 分）

阅读以下说明和 C 函数，将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

计算机在处理算术表达式时，首先将其转换为后缀表达式。例如，表达式“ $46+5*(120-37)$ ”的后缀表达式形式为“46 5 120 37 - * +”。

计算后缀表达式时，从左至右扫描后缀表达式：若遇到运算对象，则压入栈中；遇到运算符，则从栈中弹出相关运算对象进行计算，并将运算结果压入栈中，重复以上过程，直到后缀表达式扫描结束。例如，后缀表达式“46 5 120 37 - * +”的计算过程为：

- 依次将 46、5、120、37 压入栈中；
- 遇到“-”，取出 37、120，计算 $120 - 37$ ，得 83，将其压入栈中；
- 遇到“*”，取出 83、5，计算 $5*83$ ，得 415，将其压入栈中；
- 遇到“+”，取出 415、46，计算 $46+415$ ，得 461，将其压入栈中；
- 表达式结束，则计算过程完成。

函数 `computing(char expr[], int *result)` 的功能是基于栈计算后缀形式的表达式（以串形式存入字符数组 `expr`）的值，并通过参数 `result` 返回该值。函数的返回值为 -1/0 分别表示表达式有/无错误。假设表达式中仅包含数字、空格和算术运算符号，其中所有项均以空格分隔，且运算符仅包含加（“+”）、减（“-”）、乘（“*”）、除（“/”）。

函数 `computing` 中所用栈的基本操作的函数原型说明如下：

`void InitStack(STACK *s)`：初始化栈。

`void Push(STACK *s, int e)`：将一个整数压栈，栈中元素数目增 1。

`void Pop(STACK *s)`：栈顶元素出栈，栈中元素数目减 1。

`int Top(STACK s)`：返回非空栈的栈顶元素值，栈中元素数目不变。

`int IsEmpty(STACK s)`：若 `s` 是空栈，则返回 1 否则返回 0。

[C 函数]

```
int computing(char expr[], int *result)
{
    STACK s;    int tnum, a, b;    char *ptr;
    InitStack(&s);
    ptr = expr;                                /*字符指针指向后缀表达式串的第一个字符*/
    while (*ptr!='\0') {
        if (*ptr==' ') {                        /*当前字符是空格*/
                (1)    ;                        /*字符指针指向下一字符*/
            continue;
        }
        else
```



```

if (isdigit(*ptr)) {
    /*当前字符是数字, 则将该数字开始的数字串转换为数值*/
    tnum = (2);
    while (*ptr>='0' && *ptr <='9') {
        tnum = tnum * 10 + (3);
        ptr++;
    }
    Push((4));
}
else /*当前字符是运算符或其他符号*/
if (*ptr=='+' || *ptr=='-' || *ptr=='*' || *ptr=='/') {
    if (!IsEmpty(s)) {
        a = Top(s); Pop(&s); /*取运算符的第二个运算数*/
        if (!IsEmpty(s)) {
            b = Top(s); Pop(&s); /*取运算符的第一个运算数*/
        }
        else return -1;
    }
    else return -1;
    switch (*ptr) {
        case '+': Push(&s, b+a); break;
        case '-': Push(&s, b-a); break;
        case '*': Push(&s, b*a); break;
        case '/': Push(&s, b/a); break;
    }
}
else
    return -1;
ptr++; /*字符指针指向下一字符*/
} /* while */
if (IsEmpty(s)) return -1;
else {
    (5) = Top(s); Pop(&s); /*取运算结果*/
    if (!IsEmpty(s)) return -1;
    return 0;
}
}

```

从下列 3 道试题（试题五至试题七）中任选 1 道解答。如
果解答的试题数超过 1 道，则题号小的 1 道解答有效。

试题五（共 15 分）

阅读下列说明、图和 C++代码，将应填入 （n） 处的字句写在答题纸的对应栏内。

【说明】

已知对某载客车辆(Car)进行类建模，如图 5-1 所示，其中类 Engine 表示发动机引擎，类 Wheel 表示车轮，类 Body 表示车身，类 Driver 表示司机，类 Passenger 表示乘客。

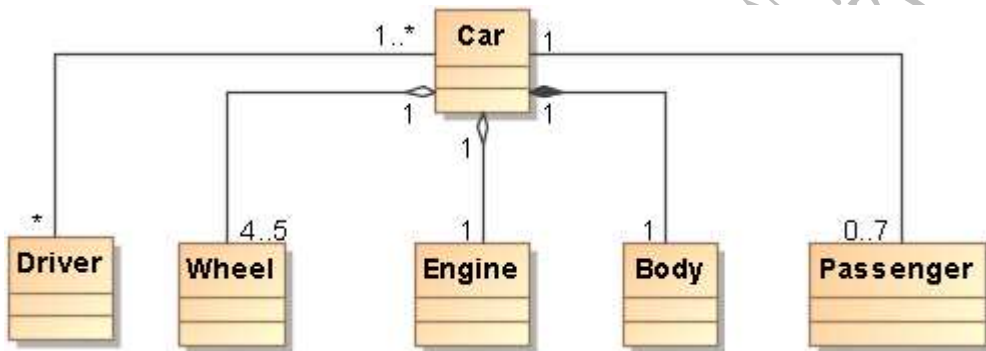


图 5-1 类图

【C++代码】

```
const int (1) = 7;           //定义最多载客数
const int MAX_WHEELS = 5;     //定义最多轮胎数

class Body{ //此处代码省略 }; //车身类
class Passenger{ //此处代码省略 }; //乘客类
class Wheel{ //此处代码省略 }; //车轮类

class Driver{ //司机类
public:
    string name; //表示第几路公交车司机
    Driver(string driverName):name((2)) {}; //构造函数
};

class Engine{ //引擎类
public:
    string engineNo; //引擎编号
    Engine(string engineNo){ (3) ->engineNo = engineNo; } //构造函数
};
```

```

class Car{ //汽车类
protected:
    Engine * engine;    Driver * driver;    Body body;
    Wheel * wheels[MAX_WHEELS];    Passenger * passengers[MAX_PASSENGERS];
public:
    Car(Driver *driver){ //构造函数
        this->driver = driver;
        engine = new Engine("TX6536 型号引擎");
        for (int index = 0; index < MAX_WHEELS; index++){
            wheels[index] = new Wheel();
        }
        for (int index = 0; index < MAX_PASSENGERS; index++){
            passengers[index] = NULL;
        }
    }
    virtual ~Car(){ //析构函数
        for (int index=0; index < MAX_WHEELS; index++){
            delete wheels[index];
        }
        delete (4);
    }
    int getPassengerNumber(){ //获取车上乘客数量
        //此处代码省略
    }
    void getOnPassenger(Passenger * aPassenger ){ //乘客上车
        //此处代码省略
    }
    void run(){ //开车
        if(driver == NULL){ cout << "司机尚未上车!";    return; }
        //此处代码省略
    }
};

void main(){
    Driver driver("第五路公交车司机");
    Car car((5));
    Passenger passengers[MAX_PASSENGERS];
    for (int index = 0 ; index < MAX_PASSENGERS; index ++){ //乘客上车处理
        car.getOnPassenger(&passengers[index]);
        car.run();
    }
}

```

试题六（共 15 分）

阅读以下应用说明以及 Visual Basic 程序代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

[应用说明]

某应用程序可选择打开用户指定的文本文件，将其内容显示在指定的文本框内供用户编辑，并将编辑后的结果保存在用户指定的文件中。运行时的窗口如图 6-1 所示，其中有六个标签、一个驱动器列表框、一个目录列表框、一个文件列表框、一个文件类型组合框、一个文件编辑文本框、一个文件名文本框以及两个命令按钮。

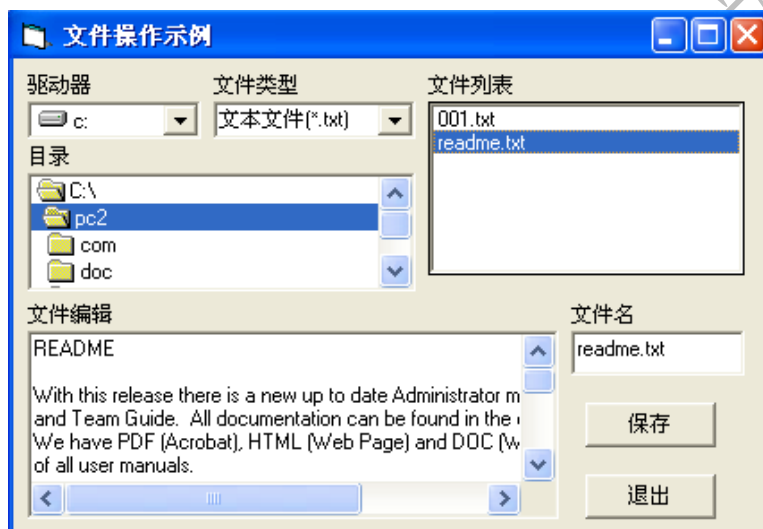


图 6-1

该程序的开发要求如下：

- (1) 通过驱动器列表框 (Drive1)、目录列表框 (Dir1) 和文件列表框 (File1)，选择文件。
- (2) 文件类型组合框 (Cmb_type) 设置为下拉式列表框，其中有三个供选项，分别为“所有文件 (*.*)”、“文本文件 (*.txt)”和“可执行文件 (*.exe)”。在文件列表框中列出的文件类型会自动与文件类型组合框中选择的文件类型相匹配。
- (3) 在文件列表框中单击一个文件名时，该文件名会显示在文件名文本框 (Txt_filename) 中。
- (4) 在文件列表框中双击一个文件名时，若是文本文件，则在文件编辑文本框 (Txt_file) 中显示该文件的内容并可进行编辑；若不是文本文件，则弹出一个对话框，提示“请选择文本文件！”
- (5) 对于编辑后的文本文件，可在文件名文本框 (Txt_filename) 中输入新的文件名，并单击命令按钮 (Cmd_save) 进行保存。

[Visual Basic 程序代码]

```
Private Sub Form_Load()  
    Cmb_type.AddItem "所有文件 (*.*)"   
    Cmb_type.AddItem "文本文件 (*.txt)"
```

```

    Cmb_type.AddItem "可执行文件(*.exe) "
    Cmb_type.ListIndex = 0
    File1.Pattern = "*. *": Txt_filename.Text = ""
    Txt_file.Text = ""
End Sub
Private Sub Dir1_Change()
    File1.Path = ____ (1) ____
End Sub
Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

Private Sub Cmb_type_click()
    Select Case Cmb_type. ____ (2) ____
        Case 0
            File1.Pattern = "*. *"
        Case 1
            File1.Pattern = "*.txt"
        Case 2
            File1.Pattern = "*.exe"
    End Select
End Sub
Private Sub Cmd_save_Click()
    usrFile = GetFileName() '函数 GetFileName 获得要保存的文件名
    Open usrFile For Output As #1 '定义 usrFile 为 1 号输出文件
    Print #1, Txt_file.Text '输出到 1 号文件
    Close #1
End Sub
Private Sub File1_Db1Click()
    If right(File1.FileName, 3) <> ____ (3) ____ Then
        MsgBox "请选择文本文件!"
        Exit Sub
    End If
    usrFile = GetFileName() '函数 GetFileName 获得要打开的文件名
    Open usrFile For Input As #1 '定义 usrFile 为 1 号输入文件
    Txt_file.Text = ""
    Do While ____ (4) ____ EOF(1)
        Line Input #1, fContext '从 1 号文件读入一行
        Txt_file.Text = Txt_file.Text + ____ (5) ____ + vbCrLf
    Loop
    Close #1
End Sub
'其他代码略

```

试题七（共 15 分）

阅读下列说明、图和 Java 代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

[说明]

已知对某载客车辆(Car)进行类建模，如图 7-1 所示，其中类 Engine 表示发动机引擎，类 Wheel 表示车轮，类 Body 表示车身，类 Driver 表示司机，类 Passenger 表示乘客。

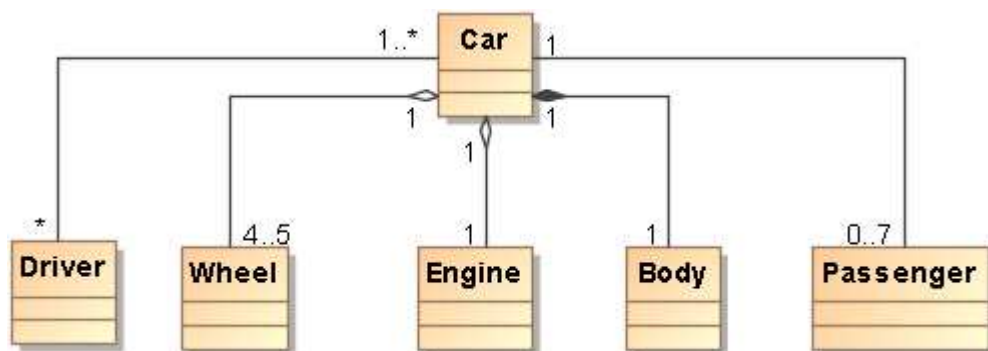


图 7-1 类图

[Java 代码]

```
class Body{           //此处代码省略    }; //车身类
class Passenger{      //此处代码省略    }; //乘客类
class Wheel{          //此处代码省略    }; //车轮类

class Driver{ //司机类
    public String name; //表示第几路公交车司机
    public Driver(String driverName){name = driverName;} //构造函数
};

class Engine{ //引擎类
    public String engineNo; //引擎编号
    public Engine(String engineNo){ this.engineNo = engineNo; } //构造函数
};

public class Car{ //汽车类
    static final int   (1)   = 7; //定义最多载客数
    static final int MAX_WHEELS = 5; //定义最多轮胎数
    protected Engine engine;
    protected Driver driver;
    protected Body body = new Body();
    protected Wheel[] wheels;
```

```

protected Passenger[] passengers;
public Car(Driver driver) { //构造函数
    (2).driver = driver;
    engine = new Engine("TX6536 型号引擎");
    wheels = new Wheel[MAX_WHEELS];
    passengers = new Passenger[MAX_PASSENGERS];
    for (int index = 0; index < MAX_WHEELS; index++) {
        wheels[index] = new Wheel();
    }
    for (int index = 0; index < MAX_PASSENGERS; index++) {
        passengers[index] = null;
    }
}

int getPassengerNumber() { //获取车上乘客数量
    //此处代码省略
}

void getOnPassenger(Passenger aPassenger ) { //乘客上车
    //此处代码省略
}

void run() { //开车
    if ( (3) ) { System.out.println("司机尚未上车!"); return;}
    //此处代码省略
}

public static void main(String args[]) {
    Driver driver = new Driver("第五路公交车司机");
    Car car = new Car ( (4) );
    for (int index = 0 ; index < MAX_PASSENGERS; index ++ )
        car.getOnPassenger ( (5) Passenger());
    car.run();
}
}

```