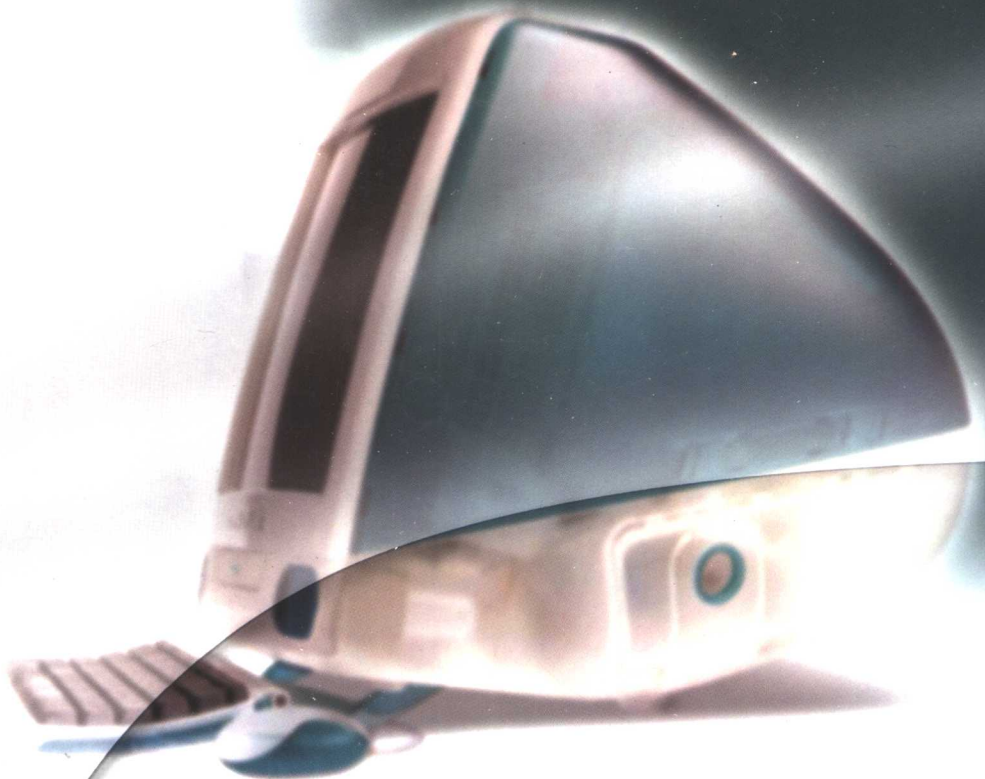


二级辅导

全国计算机等级考试
National Computer Rank Examination

基础知识和C++语言程序设计

李 季 黄志雄 编著



南开大学出版社

全国计算机等级考试系列

二级辅导

基础知识和 C++ 语言程序设计

李 季 黄志雄 编著

南开大学出版社
天 津

图书在版编目(CIP)数据

全国计算机等级考试二级辅导. 基础知识和 C++ 语言程序设计 / 李季, 黄志雄编著. —天津: 南开大学出版社, 2004. 10

(全国计算机等级考试系列丛书)

ISBN 7-310-02147-9

I. 全... II. ①李... ②黄... III. ①电子计算机—水平考试—自学参考资料②C 语言—程序设计—水平考试—自学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字(2004)第 069809 号

版权所有 翻印必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022)23508339 23500755

营销部传真: (022)23508542 邮购部电话: (022)23502200

*

天津蓟县宏图印务有限公司印刷

全国各地新华书店经销

*

2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

787×1092 毫米 16 开本 17.5 印张 444 千字

定价: 25.00 元

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022)23507125

前 言

1994 年, 国家教育部推出了面向社会的全国计算机等级考试。10 年来, 为了促进计算机知识的普及和计算机应用技术的推广, 满足用人单位考核工作人员应用计算机的水平, 全国计算机等级考试也在不断完善和修订。全国计算机等级考试最新版考试大纲在 2004 年开始推行。新考纲中, 除了取消过时的考试科目和调整一些科目的考试内容之外, 在二级考试科目中, 新增了目前较为流行的计算机语言 C++、Java 和 Access。

为了满足考生的要求, 在全国计算机等级考试中心有关专家的指导下, 我们严格按照新考试大纲的要求, 组织了既熟悉等级考试又精通专业技术的强大编写队伍, 专门针对新增的 C++ 考试科目编写了本书。

本书针对准备参加全国计算机等级考试二级 C++ 语言程序设计的考生。简明扼要的学习要点说明, 大量的例题精解, 帮助考生快速复习和熟练运用 C++ 语言的设计方法, 提高学习效率, 在较短的时间内轻松通过考试。

考试基本要求

全国计算机等级考试二级 (C++ 语言程序设计) 分为上机考试和笔试两部分, 主要内容涉及公共基础知识和 C++ 语言程序设计。

公共基础知识部分对于考生的要求是: 掌握算法的基本概念; 掌握基本数据结构及其操作; 掌握基本排序和查找算法; 掌握逐步求精的结构化程序设计方法; 掌握软件工程的基本方法, 具有初步应用相关技术进行软件开发的能力; 掌握数据库的基本知识, 了解关系数据库的设计。

C++ 语言程序设计部分对于考生的基本要求是: 掌握 C++ 语言的基本语法规则; 熟练掌握有关类与对象的相关知识; 能够阅读和分析 C++ 程序; 能够采用面向对象的编程思路和方法编写应用程序; 能熟练使用 Visual C++ 6.0 集成开发环境编写和调试程序。

笔试时间为 90 分钟, 满分 100 分 (公共基础知识的考试方式为笔试, 与 C++ 语言程序设计的笔试部分合为一张试卷, 占 30 分); 上机考试 90 分钟, 满分 100 分。

本书的特点

专家构成的作者队伍

本书由长期从事全国计算机等级考试授课、辅导和 C++ 语言程序设计的专家共同编写, 并得到考试中心有关专家的详细指点, 内容紧扣考试大纲, 书中的习题模拟考试题, 避免读者走弯路, 提高学习效率。

章节内容精心安排

本书各章都分为如下几个部分：

- **总体要求** 在每章的开始部分，列出了针对本章内容的总体要求，指出了具体应当掌握的知识。
- **学习要点和特别提示** 在随后的各小节中，“学习要点”中逐项列出与考纲内容相关的知识点，读者可以通过这部分内容复习与考试相关的知识。其间穿插的“特别提示”，针对学习要点说明需要特别注意的地方以及技巧、警告或提示信息。
- **例题精解** 在例题精解部分，针对大量的典型例题进行分析，搭起一座由教程到考试试题的桥梁，帮助考生进行自学复习，从而降低备考难度。

本书作者

本书主要由李季、黄志雄主笔，另外，在编写过程中，邱代燕、宫晓琳、高磊、董岚枫、徐增辉、刘朋、刘刚、尚卫平、孙宏、赵成璧、于樊鹏、陈河南、贺军、梁德成、梁彩隆、廖明武、倪永智、贺民、李志云、戴军、陈安南、李晓春、王春桥、王雷、郭涛、韦笑、龚亚萍、孟丽艳、石丽霞等人，在资料整理、查错、代码调试等方面做了大量的工作。

读者若对本书有任何意见和建议，均可给如下地址发邮件：

xiaoxiang-007@sohu.com

目 录

第一部分 公共基础知识

第 1 章 基本数据结构与算法	1
1.1 总体要求	1
1.2 基本数据结构	1
1.3 算法及其评价	2
1.4 线性表	2
1.5 栈和队列	2
1.6 链表	3
1.7 树.....	3
1.8 查找和排序	3
1.9 例题精解	4
第 2 章 程序设计基础	8
2.1 总体要求	8
2.2 程序设计	8
2.3 结构化程序设计	8
2.4 面向对象的程序设计方法	8
2.5 例题精解	9
第 3 章 软件工程基础	12
3.1 总体要求	12
3.2 软件工程概述	12
3.3 结构化分析与设计	12
3.4 软件测试与程序调试	13
3.5 例题精解	13
第 4 章 数据库设计基础	15
4.1 总体要求	15
4.2 数据库与数据模型	15
4.3 数据库设计	16
4.4 例题精解	16

第二部分 C++语言程序设计

第 5 章 C++概述	19
5.1 总体要求	19
5.2 程序设计语言和程序设计方法	19
5.3 了解 C++语言	20

5.4 C++程序结构	21
5.5 C++程序的开发过程	22
5.6 C++的字符集及词法约定	22
5.7 例题精解	24
第6章 C++数据类型	28
6.1 总体要求	28
6.2 基本数据类型	28
6.3 常量和变量	29
6.4 运算符和表达式	31
6.5 数据类型转换	33
6.6 复合数据类型	34
6.7 例题精解	36
第7章 控制语句	50
7.1 总体要求	50
7.2 C++语言的语句	50
7.3 顺序控制语句	50
7.4 选择控制语句	51
7.5 循环控制语句	53
7.6 跳转语句	54
7.7 例题精解	55
第8章 函数	69
8.1 总体要求	69
8.2 函数概述	69
8.3 函数的定义和调用	70
8.4 函数的参数传递	72
8.5 内联函数	73
8.6 递归函数	73
8.7 函数重载	74
8.8 作用域	74
8.9 文件与预处理	76
8.10 例题精解	78
第9章 数组和字符串	97
9.1 总体要求	97
9.2 数组	97
9.3 字符串和字符数组	99
9.4 常用的字符串函数	100
9.5 例题精解	101
第10章 指针与引用	122
10.1 总体要求	122

10.2	指针	122
10.3	指针与函数	124
10.4	指针与数组	125
10.5	引用	126
10.6	例题精解	127
第 11 章	类与对象	153
11.1	总体要求	153
11.2	类	153
11.3	类对象	154
11.4	构造函数和析构函数	156
11.5	静态成员	157
11.6	常类型	158
11.7	对象指针和对象引用	158
11.8	this 指针	159
11.9	友元	159
11.10	例题精解	160
第 12 章	派生与继承	185
12.1	总体要求	185
12.2	基类和派生类	185
12.3	访问控制	186
12.4	派生类的构造函数和析构函数	187
12.5	虚基类	188
12.6	例题精解	189
第 13 章	多态性与虚函数	208
13.1	总体要求	208
13.2	静态联编和动态联编	208
13.3	虚函数	209
13.4	纯虚函数和抽象类	210
13.5	运算符重载	210
13.6	例题精解	211
第 14 章	模板	230
14.1	总体要求	230
14.2	模板的概念	230
14.3	函数模板	231
14.4	类模板	231
14.5	例题精解	232
第 15 章	C++流和文件流	238
15.1	总体要求	238
15.2	C++流	238

15.3 格式化 I/O	242
15.4 文件流	244
15.5 例题精解	246
第 16 章 上机指导	262
16.1 总体要求	262
16.2 例题精解	262
附录 C++ 运算符优先级表	270

第一部分 公共基础知识

第1章 基本数据结构与算法

本章着重讲解了数据结构及相关的基本概念、几种典型的数据结构及其操作；算法的概念以及算法复杂度；主要的查找及排序算法。

1.1 总体要求

【具体要求】

- 算法的基本概念；算法复杂度的概念和意义。
- 数据结构的定义；数据的逻辑结构与存储结构；数据结构的图形表示；线性结构与非线性结构的概念。
- 线性表的定义；线性表的顺序存储结构及其插入与删除运算。
- 栈和队列的定义；栈和队列的顺序存储结构及其基本运算。
- 线性单链表、双向链表与循环链表的结构及其基本运算。
- 树的基本概念；二叉树的定义及其存储结构；二叉树的前序、中序和后序遍历。
- 顺序查找与二分查找算法；基本排序算法（交换类排序，选择类排序，插入类排序）。

1.2 基本数据结构

【学习要点】

1. 数据结构的基本概念及内容。
2. 线性结构的逻辑特征与非线性结构的逻辑特征。
3. 数据存储结构的四种基本方式：顺序存储、链式存储、索引存储、散列存储。
4. 数据的运算：查找运算、读取运算、插入运算、删除运算、更新运算。
5. 不同类型数据结构如何用图形表示。

【特别提示】

注意区分线性结构与非线性结构的不同。

表 1.1 线性结构与非线性结构比较

线性结构	非线性结构（以树为例）
第一个数据元素——无前驱	根结点——无前驱
最后一个数据元素——无后继	多个叶子结点——无后继
其他数据元素——一个前驱，一个后继	其他数据元素——一个前驱，多个后继

1.3 算法及其评价

【学习要点】

1. 算法的基本概念与特性。
2. 评价一个算法优劣的四个方面：正确性、运行时间、占用空间和简单性。其中主要的是算法的运行时间和占用空间。
3. 算法的时间复杂度和空间复杂度。

【特别提示】

一般地，时间复杂度有如下关系：

$$O(1) \leq O(\log_2 n) \leq O(n) \leq O(n \log_2 n) \leq O(n^2) \leq O(n^3) \leq \dots \leq O(2^n)。$$

1.4 线性表

【学习要点】

1. 线性表的基本概念。
2. 线性表的顺序存储结构。
3. 线性表的插入和删除运算。

【特别提示】

在顺序表中，每个结点 a_i 的存储地址是该结点在表中的位置 i 的线性函数，只要知道基地址和每个结点的大小，就可在相同时间内求出任一结点的存储地址。因此顺序表是一种随机存储结构。

假设表中每个结点占用 c 个存储单元，并设表中开始结点 a_1 的存储地址（简称为基地址）是 $LOC(a_1)$ ，那么结点 a_i 的存储地址 $LOC(a_i)$ 可通过下式计算：

$$LOC(a_i) = LOC(a_1) + (i-1) * c \quad (1 \leq i \leq n)。$$

1.5 栈和队列

【学习要点】

1. 栈的基本概念。

2. 栈的顺序存储结构与运算。
3. 队列的顺序存储结构与运算。

【特别提示】

注意栈和队列在操作上的区别。

1.6 链表

【学习要点】

1. 单链表存储结构的特点。
2. 循环链表存储结构的特点。
3. 双向链表存储结构的特点。
4. 链表的基本操作：插入和删除。

【特别提示】

循环链表的结构与普通链表一样，只是普通链表的最后一个结点的指针域取值为空，而循环链表的最后一个结点的指针域指向第一个结点。

1.7 树

【学习要点】

1. 树以及树的相关概念。
2. 二叉树及其相关概念，二叉树的顺序存储结构和链式存储结构。
3. 二叉树的先序、中序、后序遍历。

1.8 查找和排序

【学习要点】

1. 查找的基本概念。
2. 常用的查找方法：顺序查找和二分查找。
3. 排序的基本概念。
4. 各种排序方法的原理和实现。
5. 各种排序法的比较：

(1) 直接插入排序算法的时间复杂度是 $O(n^2)$ ，它是一种稳定的排序方法。

(2) 希尔排序算法的时间复杂度是 $O(n\log_2 n)$ ，它的实质是一种插入排序，但它是一种不稳定的排序方法。

- (3) 冒泡排序算法的时间复杂度是 $O(n^2)$ ，它是一种稳定的排序方法。
- (4) 快速排序算法的时间复杂度是 $O(n\log_2 n)$ ，它是一种不稳定的排序方法。
- (5) 直接选择排序算法的时间复杂度是 $O(n^2)$ ，它是一种不稳定的排序方法。
- (6) 堆排序算法的时间复杂度是 $O(n\log_2 n)$ ，它是一种不稳定的排序方法。
- (7) 归并排序算法的时间复杂度是 $O(n\log_2 n)$ ，它是一种稳定的排序方法。
- (8) 基数排序算法的时间复杂度是 $O(d*(rd+n))$ ，其中 rd 是基数， d 是关键字的位数， n 是元素个数，它是一种稳定的排序方法。

1.9 例题精解

【例 1.1】在数据结构中，从逻辑上可以把数据结构分成（ ）。

- A) 动态结构和静态结构 B) 线性结构和非线性结构
C) 动态结构和线性结构 D) 静态结构和非线性结构

【解】本题考核数据的逻辑结构。逻辑结构即数据元素之间的逻辑关系。它可以用一个数据元素的集合和定义在此集合上的若干关系来表示。值得注意的是，数据的逻辑结构是从逻辑关系上描述数据，与数据的存储无关，是独立于计算机的。可分为两大类：线性结构和非线性结构。

本题的答案是 B。

【例 1.2】线性表的链式存储结构是一种（ ）的存储结构。

- A) 随机存储 B) 顺序存储
C) 索引存储 D) 散列存储

【解】本题考核数据的存储结构。对于顺序存储方式，每一个存储结点只含一个数据元素，所有的存储结点相继存储在一个连续的存储区里，用存储结点之间的位置关系表示数据元素之间的逻辑关系，因此属于随机存储。而链式存储方式的每一个存储结点不仅含有一个数据元素，还包括指针，每一个指针指向一个与本结点有逻辑关系的结点，因此属于顺序存储。

本题的答案是 B。

【例 1.3】下面这个程序段的时间复杂度是（ ）。

```
for (i=1; i<n; i++)
{
    y=y+1;                               ①
    for (j=0; j<=(2*n); j++)
        x++;                             ②
}
```

- A) $O(\log_2 n)$ B) $O(n)$
C) $O(n\log_2 n)$ D) $O(n^2)$

【解】本题考核算法时间复杂度的计算。语句的频度指的是该语句重复执行的次数。一个算法中所有语句的频度之和构成了该算法的运行时间。在本例算法中，其中语句①的

频度是 $n-1$ ，语句②的频度是 $(n-1)(2n+1)=2n^2-n-1$ ，则该程序段的时间复杂度是 $T(n)=n-1+2n^2-n-1=O(n^2)$ 。

本题答案是 D。

【例 1.4】下面这个程序段的时间复杂度是 ()。

$i=1;$ ①

while ($i \leq n$)

$i=i*2;$ ②

A) $O(\log_2 n)$

B) $O(n)$

C) $O(n \log_2 n)$

D) $O(n^2)$

【解】本题考核算法时间复杂度的计算。其中语句①的频度是 1，设语句②的频度是 $f(n)$ ，则有 $2^{f(n)} \leq n$ ，即 $f(n) \leq \log_2 n$ ，取最大值 $f(n) = \log_2 n$ 。则该程序段的时间复杂度 $T(n) = 1 + f(n) = 1 + \log_2 n = O(\log_2 n)$ 。

本题答案是 A。

【例 1.5】一个向量第一个元素的存储地址是 100，每个元素的长度为 2，则第 5 个元素的地址是 ()。

A) 110

B) 108

C) 100

D) 120

【解】本题考核数据元素存储地址的计算。第 5 个元素的地址 $= 100 + 2 * (5-1) = 108$ 。本题答案是 B。

【例 1.6】一个栈的入栈序列是 1, 2, 3, ..., n ，其输出序列为 $P^1, P^2, P^3, \dots, P^n$ ，若 $P^1 = n$ ，则 P^i 为 ()。

A) i

B) $n=i$

C) $n-i+1$

D) 不确定

【解】本题考核栈的基本操作。当 $P^1 = n$ ，即 n 是最先出栈的，根据栈的运算原理， n 必定是最后入栈的，那么输入顺序必定是 1, 2, 3, ..., n ，则出栈的序列是 $n, n-1, n-2, \dots, 1$ ，所以答案是 C。

本题答案 C。

【例 1.7】如图 1.1 所示的二叉树的先序遍历序列是_____，中序遍历序列是_____，后序遍历序列是_____。

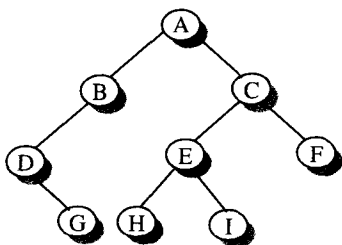


图 1.1 二叉树

【解】本题考核二叉树的先序、中序、后序遍历操作。注意三种遍历方法的不同：先序遍历为 (1) 访问根结点 (2) 先序遍历左子树 (3) 先序遍历右子树；中序遍历为 (1)

中序遍历左子树 (2) 访问根结点 (3) 中序遍历右子树; 后续遍历为 (1) 后序遍历左子树 (2) 后序遍历右子树 (3) 访问根结点。按照定义, 先序遍历序列是 A B D G C E H I F, 中序遍历序列是 D G B A H E I C F, 后序遍历序列是 G D B H I E F C A。

本题答案是: A B D G C E H I F; D G B A H E I C F; G D B H I E F C A。

【例 1.8】假设要在链表的两个数据元素 a 和 b 之间插入一个数据元素 x, 已知 p 为其链表存储结构中指向结点 a 的指针。如图 1.2 所示。为插入数据元素 x, 首先要生成一个数据域为 x 的结点, 然后插入在链表中, 即修改结点 x 中的指针域, 令其____①____, 然后修改结点 a 中的指针域, 令其____②____, 从而实现三个元素 a、b 和 x 之间逻辑关系变化。

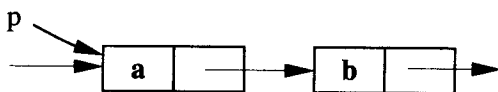


图 1.2 链表结构示意图

【解】本题考核链表的插入操作。根据插入操作的逻辑定义, 第一步, 生成一个数据域为 x 的结点, 准备插入在链表中; 第二步修改结点 x 中的指针域, 令其指向结点 b; 第三步是使结点 a 中的指针域指向结点 x。从而实现元素 x 的插入。插入后的链表如图 1.3。

本题答案是: ①指向结点 b; ②指向结点 x。

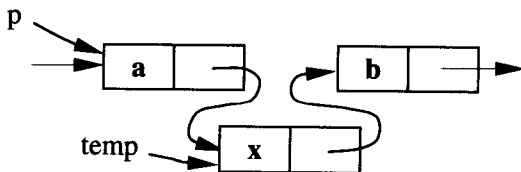


图 1.3 在链表中插入 X 的示意图

【例 1.9】已知序列{17,18,60,40,7,32,73,65,85}, 采用冒泡排序法对该序列作降序排序时, 第四趟的结果是_____。

【解】本题考核冒泡排序法。冒泡排序法的算法基本思想是: 通过无序区中相邻记录关键字间的比较和位置交换, 使关键字最小的记录如气泡一般逐渐往上“漂浮”直至“水面”。整个算法是从最下面的记录开始, 对每两个相邻的关键字进行比较, 且使关键字较小的记录换至关键字较大的记录之前, 使得经过一趟冒泡排序后, 关键字最小的记录达到最上端。接着, 再在剩下的记录中找关键字最小的记录, 并把它换在第二个位置上。依此类推, 一直到所有记录都有序为止。根据这种思想, 每趟的排序结果如下:

初始: 17, 18, 60, 40, 7, 32, 73, 65, 85

第一趟 18, 60, 40, 17, 32, 73, 65, 85, 7

第二趟 60, 40, 18, 32, 73, 65, 85, 17, 7

第三趟 60, 40, 32, 73, 65, 85, 18, 17, 7

第四趟 60, 40, 73, 65, 85, 32, 18, 17, 7

第五趟 60, 73, 65, 85, 40, 32, 18, 17, 7

第六趟 73, 65, 85, 60, 40, 32, 18, 17, 7

第七趟 73, 85, 65, 60, 40, 32, 18, 17, 7

第八趟 85, 73, 65, 60, 40, 32, 18, 17, 7

排序结束。

本题答案是: 60, 40, 73, 65, 85, 32, 18, 17, 7。

【例 1.10】已知序列{503, 17, 512, 908, 170, 897, 275, 653, 426, 154, 509, 612, 677, 765, 703, 94}, 采用希尔排序法 ($d_1=8$) 对该序列作升序排序, 第三趟的 d 值为 ①, 第三趟的排序结果是 ②。

【解】本题考核希尔排序法。希尔排序法的算法的基本思想是: 把记录按下标的一定增量分组, 对每组记录使用插入排序, 随着增量逐渐减小, 所分成的组包含的记录越来越多, 到增量的值减小到 1 时, 整个数据合为一组, 构成一组有序记录, 则完成排序。根据这种思想, 每趟的排序结果如下:

初始: 503, 17, 512, 908, 170, 897, 275, 653, 426, 154, 509, 612, 677, 765, 703, 94

第一趟 ($d_1=8$) 426, 17, 509, 612, 170, 765, 275, 94, 503, 154, 512, 908, 677, 897, 703, 653

第二趟 ($d_2=4$) 170, 17, 275, 94, 426, 154, 509, 612, 503, 765, 512, 653, 677, 897, 703, 908

第三趟 ($d_3=2$) 170, 17, 275, 94, 426, 154, 503, 612, 509, 653, 512, 765, 677, 897, 703, 908

第四趟 ($d_4=1$) 11, 94, 154, 170, 275, 426, 503, 509, 512, 612, 653, 677, 703, 765, 897, 908

排序结束。

本题答案是: ①2; ②170, 17, 275, 94, 426, 154, 503, 612, 509, 653, 512, 765, 677, 897, 703, 908。

第2章 程序设计基础

本章讲解了程序设计方法及程序设计的风格、原则；结构化程序设计；面向对象程序设计的思想及相关的基本概念；面向对象程序设计的方法以及采用面向对象方法的优点。

2.1 总体要求

【具体要求】

- 程序设计方法与风格。
- 结构化程序设计。
- 面向对象的程序设计方法，掌握理解对象、方法、属性以及继承与多态性的概念。

2.2 程序设计

【学习要点】

1. 程序设计的基本方法。
2. 程序设计的风格，包括源程序文档化、数据说明、语句结构、输入 / 输出方法。

【特别提示】

程序设计的风格非常重要。养成良好的程序设计风格，既有助于别人阅读我们写好的程序，又方便编程人员自己后期调试程序，可节省很多时间，降低了程序出错的可能性。

2.3 结构化程序设计

【学习要点】

1. 结构化程序设计的基本特征。
2. 结构化程序设计的主要原则。

2.4 面向对象的程序设计方法

1. 什么是面向对象的程序设计。
2. 面向对象方法的主要特点。