天勤论坛

天勤计算机考研系列

杭电刘春英 鼎力推荐!

计算机考研复试上机指导全书

JISUANJI KAOYAN FUSHI SHANGJI ZHIDAO QUANSHU

ACM俱乐部 天勤论坛 组编
孙肇博 张飞飞 主编

复试宝典

▲ 复试上机资料查询微信公众平台

此微信公众平台收录了大量的复试必备资料及面试技巧。

天勤论坛微信二维码

更多计算机考研和学习交流尽在www.csbiji.com

天勤论坛

天勤论坛，取名自古训“天道酬勤”，意为考研路上，困苦实多，然而天自有道，勤恳付出者，必有应得之酬劳。

由天勤论坛组编的高分笔记系列计算机考研辅导书，融入了论坛答疑的精华内容，论坛组织了高分考生进行勘误，不断完善此套书籍。考生在书中遇到疑问，也可在线与作者进行交流。

机械工业出版社
CHINA MACHINE PRESS

天勤计算机考研系列

计算机考研复试上机指导全书

ACM 俱乐部　天勤论坛　组编
孙肇博　张飞飞　主编

机 械 工 业 出 版 社

本书作者针对2000年以来若干所名校的计算机专业研究生入学考试复试上机考试真题进行了深入的分析解读，选取了若干道具有代表性的真题并按照知识点将其分类，以一种启发式的讲解来引导考生该如何解决上机考试中的问题；以独特的视角解答考生对于复试上机考试整个过程的各种疑问，缓解考生的备考压力；以全面的分析和亲身经历给考生指引一条高效的复习道路。考生如果对书中的任何地方有疑问都可以与作者进行在线互动，作者会帮助考生及时解决复习中遇到的疑难点，并且考生可在配套的在线系统（zju.acmclub.com）进行练习，最大程度地提高复习效果。

本书可作为参加计算机专业研究生入学考试复试上机考试的复习指导用书，也可作为计算机专业的学生或对编程感兴趣的人员学习编程的辅导用书。

责任编辑邮箱：jinacmp@163.com。

图书在版编目（CIP）数据

计算机考研复试上机指导全书/孙肇博，张飞飞主编. —北京：机械工业出版社，2014.1

（天勤计算机考研系列）

ISBN 978-7-111-45395-6

Ⅰ.①计… Ⅱ.①孙…②张… Ⅲ.①电子计算机-研究生-入学考试-自学参考资料 Ⅳ.①TP3

中国版本图书馆CIP数据核字（2013）第319806号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）

策划编辑：吉 玲 责任编辑：吉 玲 吴超莉 任正一

封面设计：张 静 责任印制：杨 曦

北京中兴印刷有限公司印刷

2014年1月第1版第1次印刷

184mm×260mm · 7.25印张 · 182千字

标准书号：ISBN 978-7-111-45395-6

定价：17.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务 网络服务

社服务中心：(010)88361066 教材网：http://www.cmpedu.com

销售一部：(010)68326294 机工官网：http://www.cmpbook.com

销售二部：(010)88379649 机工官博：http://weibo.com/cmp1952

读者购书热线：(010)88379203 **封面无防伪标均为盗版**

序

如今，IT 行业迅猛发展，越来越多的学生希望投身于 IT 行业来发挥自己的才能，实现自己的梦想。正因如此，计算机专业一直是硕士研究生入学考试中的热门专业。然而，许多前期付出了艰辛努力来通过初试的考生，最终却被复试挡在了录取大门之外，这不由让我们感到惋惜。究其原因，很多考生较弱的上机考试成绩直接导致了自己被淘汰的悲剧。本书的出版会给有志于攻读计算机专业研究生的考生们带来很大的帮助，它将指导考生们如何有针对性地备考研究生复试上机考试。

本书的作者结合自身多年的 ACM 竞赛经验，在深入分析了近年来若干所名校的计算机专业复试上机真题之后，为考生们编写出了这本辅导书，本书特点为：

（1）内容全面。书中不仅有上机考试真题的解析，还有一些对考生常见疑惑的解答、编程语言的基础和注意点、在线练习平台的使用方法等。可以说，本书在多个环节为考生们提供了帮助，让大家对上机考试有一个全面的把握。

（2）重点突出。本书作者对收集到的历年名校的上机考试真题进行了细致的分析和推敲，对上机考试中出现频率较高的知识点进行提炼，并且每个知识点均配有具有代表性的真题进行实战讲解，让考生们的备考更有针对性。

（3）在线练习。本书与 ACM 俱乐部这一在线练习平台充分结合，书中的所有题目均收录在浙江大学 ACM 俱乐部（zju.acmclub.com）中，考生们可以在上面进行免费的在线编程练习。除此之外，浙江大学 ACM 俱乐部中还有更多的历年上机考试真题和各种各样的编程语言、数据结构、算法练习题供考生和编程爱好者练习使用。通过这种“书本+在线练习”的模式，考生们能够更加真切地体验上机考试的环境，更加高效地备考，达到事半功倍的效果。

需要提醒的是——攻克上机考试最重要的一点就是坚持，只要坚持在线练习，再辅以本书的指导，那么，成功会离你越来越近的。

预祝各位考生取得好成绩！

刘春英（LCY）

2014 年 1 月于杭州电子科技大学

前　言

考研是一件十分艰辛的事，在经过初试的考验之后，还有复试这最后一关在等待着考生。对于计算机专业考研来说，复试尤其特别。由于计算机学科是理论与实践结合非常紧密的学科，因此各个学校尤为看重考生的实际动手编程的能力。然而在初试中，考生的实际动手编程的能力无法在纸面上得到考查。正因如此，越来越多的学校在计算机专业考研的复试中增加了上机考试，并且上机考试的分值在整个复试中所占的比例很高，有些学校更是规定复试成绩不及格者不予录取。

不幸的是，有很大一部分考生对理论知识掌握得很好，但是上机编程的经验很少，或者几乎为零。正因如此，每年都有初试分数不错的考生因为上机考试成绩很差而被淘汰的情况发生。说实话，这些因为上机考试成绩差而被淘汰的考生非常可惜。上机考试本身并没有想象中那么难，一般来说，只要备考的方法得当，在上机考试中拿到一个不错的分数要比在初试中拿到一个不错的分数简单很多。不过，正是因为没有专门的书籍或资料来指导考生该如何正确、高效地备考上机考试，反而让上机考试成为了许多考生心中一个难以逾越的坎。鉴于此种情况，作者编写了本书。在书中作者先解答了一些考生经常会感到困惑的关于上机考试的问题，为考生的备考打下一个良好的心理基础。随后，对一些在上机考试中需要经常用到和注意的编程语言方面的内容进行了讲解，并且指导考生如何运用在线判题系统来进行练习。最后，作者选择了若干道具有代表性的各个名校历年的上机考试真题，将其按照知识点分类，并以一种启发式的讲解来引导考生在拿到问题后该如何去分析解决。请记住一点，备考上机考试没有什么诀窍，成败完全取决于你平时是否真的坚持并认真练习了！

由于本书今年是第一次出版，难免会有错误或有待改进的地方，欢迎读者来 ACM 俱乐部的交流论坛（bbs.acmclub.com）指出书中的错误，或提出宝贵的建议。

本书的使用方法

本书中讲解了许多学校历年的上机考试真题，但是并没有对题目中涉及的具体算法进行过多的讲解。本书主要是告诉读者拿到问题后思考应该如何去分析，应该选用怎样的算法并通过怎样的步骤来解决问题。因此，希望读者将本书与《数据结构高分笔记》配合起来使用，当你对某个具体算法不了解时可以去那本书中找到深入浅出的讲解。同时，上机考试考的是实际动手编写代码的能力，所以只看本书而不去动手编写代码是万万不能的。为了给读者提供一个练习编程的好地方，我们特意开发了 ACM 俱乐部系统，请读者登录 zju.acmclub.com 进行注册练习。本书中的所有例题都收录在里面，同时还有本书中没有收录的各个学校历年的考研机试真题和 C 语言、数据结构算法的一些基础入门题目，每道题目都提供了参考程序和专门的讨论帖，希望读者能够多多练习！

作　者

目　录

第 3 部分 常用库函数与 STL

第 4 部分 真题中常考的数据结构与算法

第0部分　计算机专业考研复试上机考试简介

同学们应该都知道，考研初试过线并不代表你一定能被最终录取！特别是计算机专业，复试的重要性尤为突出。而在计算机专业的复试中，最重要的就是上机考试。本部分将介绍计算机复试上机考试的相关情况，力求包含考生应该知道的复试上机考试的所有要点，帮助考生做好迎战复试上机考试的心理准备。

第0章　复试上机先知道

0.1　复试上机考试的重要性

为什么现在越来越多的学校的计算机专业复试都会设置上机考试这一环节？而且有些学校明确说明了如果复试上机考试成绩零分的话，即使初试成绩再高也不予录取，这到底是为什么呢？其实答案很简单，计算机作为一门理论性和应用性都很强的学科，对学生的思维和动手能力均有较高的要求，而通过实际的上机动手编程来解答一些具有一定思维逻辑的题目正是考查学生这两方面能力的绝好手段（虽然有很多学校还没有上机考试，但是在复试的笔试中也会考查编程能力，只不过是写到纸上，没有上机考试这么直接）。而能够进入复试的同学，实际上他们在理论和思维逻辑方面的能力相差不大，特别是能够进入名校计算机专业复试的同学，可以说他们在这方面的差别就更小了。但是，能不能将自己的思路转化成正确的代码则体现了一个学生的编程功底。编程是一个熟能生巧的活，只有不断地动手练习才能将其化为自己的一种基本能力。

如果同学们多了解一些学校的计算机专业整个复试流程安排的话，你会发现上机考试都是安排在面试的前面进行，而面试时老师们都已经知道或是会问你的上机考试成绩，如果你上机考试成绩高的话，那么在面试时会非常主动，如果你的初试成绩又在中上游的话，那你基本稳录取了。这是为什么呢？其实，说的直白点，老师招学生进来读研的一个很重要的目的就是让他做项目，而做项目需要有一定的编程能力，虽然编程能力可以慢慢培养，但是老师还是喜欢能够尽快上手的学生，所以上机考试成绩高的学生在复试中很吃香。

0.2　复试上机考试流程

一般复试上机考试的流程为，先给考生纸质题目或网页题目，考生读完题后在计算机上预先安装好的编程软件（如 VC 6.0 等）中进行编码、调试，如果考生觉得自己的代码是正确的，就可以通过学校提供的系统进行提交，并且系统会实时返回对代码的评判结果，如果代码没有完全正确，则需要考生继续改正并再次提交。也有的学校是叫老师来手动输入测试数

据看输出结果是否正确。总之，大体的流程是一样的。

0.3 ACM 竞赛与复试上机考试的比较

相信很多参加计算机专业考研的同学都知道 ACM 竞赛，或者也都听说过复试上机考试的形式和 ACM 竞赛的形式一样，但是由于自己没有参加过这个竞赛，所以有一些担心。其实，这种担心完全是多余的。

先来说一下什么是 ACM 竞赛。ACM 竞赛的全称是国际大学生程序设计竞赛（ACM International Collegiate Programming Contest，ACM-ICPC），由美国计算机协会（Association for Computing Machinery，ACM）主办。这个比赛的形式是 3 人为一个团队，在 5 个小时的时间内编程解决若干个问题。计算机复试上机考试与之类似，也是在规定的时间内编程解决若干个问题，只不过是自己一个人解决。在问题的难度上，两者就相差太多了。ACM 竞赛基本上是考查一些高级算法、高级数据结构和思维逻辑很复杂的题目。而计算机复试上机考试考查的都是一些基本的算法和数据结构，或是思维逻辑比较简单的题目，当然也会有为数不多的较难的题目出现，不过放到 ACM 竞赛中顶多也只能算难度中等偏下的题目。另外，ACM 竞赛中的题目对代码的运行时间和空间占用量有着极其严格的限制，而计算机复试上机考试的题目对这两方面就没有这么苛刻，甚至有的学校复试上机考试采用按通过的测试样例个数给分（浙江大学就是这样做的），这样即使你的程序不是完全正确或没有做一定的优化，也可能会得到一定的分数。所以，同学们不用因为做不出 ACM 竞赛级别的题目，就担心上机考试没希望了，只要打好基础，在上机考试中拿到一个可观的分数还是很现实的。

说到这，可能有些同学会问那些参加过 ACM 竞赛的同学在上机考试中岂不是很有优势？没错，优势肯定是有的。但是请同学们注意，参加过 ACM 竞赛并且取得了一定成绩的同学大部分都选择直接工作或是保研了，只有很少一部分选择考研，而这其中还有一部分过不了初试的。所以在进入复试的同学中，就上机考试而言，绝大多数的同学都处于同一起跑线上。

0.4 编程语言的选择

首选 C/C++，如果考生确实对 Java 比较熟练的话，可以用 Java。但是很多学校的上机考试不让用 Java，只能用 C/C++，所以考生应该事先确认所报考的学校的复试要求。除此之外，不建议大家用别的编程语言来进行上机考试。

实际上，几乎每个学校都会指明上机考试考查 C/C++，个别的也会考查 Java，不过由于 Java 本身的特点，其在运行效率和运行空间占用量这两方面均比 C/C++落后很多（C/C++更加偏向底层，代码运行效率比 Java 高得多，内存占用量也比 Java 小得多），而上机考试的题目对代码的运行时间和空间占用量都有严格的要求，所以为了避免这个硬伤，建议大家用 C/C++。不仅如此，C/C++自带的函数库和 STL 为我们提供了大量的现成函数，我们可以充分利用这些函数来简化和优化代码。常用的库函数和 STL 会在第 3 部分讲解。

0.5 复试上机考试练习平台介绍

准备上机考试，光看书是万万不行的！自己动手码代码才是真理！但是到哪才能找到跟

上机考试环境一样的练习场所呢？不用担心，来 ACM 俱乐部的官方 OJ（zju.acmclub.com）系统练习，OJ 系统的全称是 Online Judge，是一个在线判题系统。国内有几个比较出名的 OJ 系统，如 ZOJ（acm.zju.edu.cn）、POJ（poj.org）、HDOJ（acm.hdu.edu.cn），虽然这几个 OJ 系统里的题目众多，但是基本都是为 ACM 竞赛设计的，题目整体难度要比上机考试的题目难很多，因此不建议同学们去这几个 OJ 系统备考上机考试，以免打击自己的自信心。

本书中所有的例题都收录在其中的“名校复试机考真题”这个分类中。除此之外，很多没有收录在本书中的真题也都可以在此分类中找到，每道题目都提供了专门的讨论帖并给出参考代码，希望同学们在阅读本书的时候充分利用这个平台，多多练习，这样才能使复习达到事半功倍的效果！如果你能把系统里提供的 200 余道真题都做出来，那你的上机考试肯定没问题！另外，也建议同学们做做“入门题”这个分类中的题目，可以让你得到更加全面的提高。

ACM 俱乐部的用户使用手册可以在 ACM 俱乐部论坛（bbs.acmclub.com）中的全局置顶帖中下载。

第 1 部分　程序设计基础

想要编程解决问题，最基本的当然是要掌握一门编程语言。在前面内容中指明了 C/C++在计算机复试上机考试中的优势，也建议大家用 C/C++来进行练习和考试。因此，本部分将讲解一些 C/C++的基础知识和一些在编码过程中需要注意的细节。如果你对 C/C++很熟练了，可以跳过本部分，直接往后看。

第 1 章　C/C++基础入门

1.1　头文件

首先我们来看下面的一个程序：

【代码 1-1】

```
int main() {
    int sum, a=5, b=6;
    sum=a+b;
    return 0;
}
```

这是一个编译完全正确的 C 语言程序。也许很多人会觉得奇怪，为什么没有“#include <stdio.h>”呢？也许你早就知道了，上面这行代码的作用是包含头文件。那么没有包含头文件的 C 语言程序是正确的么？我已经说了，【代码 1-1】编译完全是正确的。一个程序只要语法没有错误，编译就是正确的。同时【代码 1-1】运行也是完全正确的。我想你可能会问，那为什么在每次写 C 语言程序时几乎都包含<stdio.h>头文件？那是因为程序需要，stdio 代表的是“standard input & output”，<stdio.h>头文件的作用就是使你的程序具有输入或者输出功能。【代码 1-1】既没有输入也没有输出，因而不必包含<stdio.h>头文件。也就是说，是否要包含头文件得视你的程序需要而定。除了上面提到的<stdio.h>外，常用的头文件还有<string.h>、<math.h>、<stdlib.h>等。在第 3 部分会详细讲解这些常用头文件里的函数的用法。

那头文件到底是什么呢？一般而言，每个 C/C++程序通常由头文件（header files）和定义文件（definition files）组成（还有其他类型的文件，在此就不介绍了）。在定义文件中往往包含某个头文件。包含头文件的语法是：#include<xxx.h>。包含某个头文件往往是为了使用该头文件中声明的函数或者数据等，例如在输入和输出时为了使用 scanf 和 printf 函数，需要包含<stdio.h>头文件，为了使用字符串处理函数往往需要包含<string.h>头文件。在写代码时往往需要使用已经定义好的库函数，而使用这些函数就需要包含相应的头文件。当然，如果自己定义的函数本身就写在主函数所在的文件中时，就没有必要再添加什么头文件了。

定义文件也是一类常见的文件，主要是实现某些操作。在头文件中声明的函数往往就是在定义文件中实现的。那么头文件与定义文件有哪些共同点与不同点呢？

共同点：头文件与定义文件在文件格式上是相同的，都是文本格式。可使用任何文本编辑软件打开。

不同点：首先，文件扩展名不同。头文件以.h 为扩展名（C++中有些则以.hpp 为扩展名，这里的 p 是 plus 的缩写），而定义文件则以.c 为扩展名（C++中一般以.cpp 为扩展名）。其次，文件内容不同。往往在头文件中定义变量、数据类型以及声明函数，而在定义文件中用代码实现在头文件中声明的函数以及其他功能。

C++中的头文件都不带.h 扩展名。当然，由于 C++兼容 C 语言的语法，所以也可以使用<stdio.h>、<string.h>等头文件，不过原来 C 语言中那些带.h 扩展名的头文件在 C++中有新的名字，就是去掉.h 扩展名，最前面加一个 c，例如<cstdio>、<cstring>等。同时，在使用 C++头文件的时候记得在写完所有头文件之后写上“using namespace std;”。

1.2 数据类型

本小节将详细介绍经常用到的数据类型。首先介绍 C/C++中的基本数据类型，然后介绍一些构造数据类型、指针类型、引用类型以及空类型。

1.2.1 基本数据类型

基本数据类型最主要的特点是值不可以再分解为其他类型。有时基本数据类型也称为内置类型，而将构造类型与指针类型称为复合类型。基本类型一般分为字符、整数、浮点数。

1. C 语言中的基本数据类型

C 语言中的基本数据类型见表 1-1。

表 1-1　C 语言中的基本数据类型

大的分类	小的分类	在内存中所占的字节数/B
字符	char	1
整数	short	2
	int	2/4
	long	4
浮点数	float	4
	double	8

对于上述基本数据类型，有以下几点需要注意：

1）int 类型在内存中所占的字节数取决于编译系统。规定是 int 类型在系统中所占的字节数不小于 short 类型，不大于 long 类型，因此会有 2B 和 4B 这两种情况。

2）对于字符类型与整数类型，都有一个对应的无符号类型来表示不包含负数部分的值，从而正数的表示范围会扩大，例如 unsigned char、unsigned int 等。

3）double 类型比 float 类型精度要高，在练习时如果要定义浮点数变量，建议都定义成 double 类型，这样可以避免一些很隐晦的数据溢出的错误。

4）如果想知道某一类型或者某一变量在内存中占有的空间，可使用 sizeof 来获取，例如

sizeof(int)。

在练习的过程中可能会碰到让你处理汉字的题目，那么应该如何判断一个字符是不是汉字呢？要想知道答案，就要先知道在计算机中是如何表示汉字的。在计算机中，两个连续的字节表示一个汉字（英文字符仅用一个字节），每个字节的最高位都是 1（而英文字符的最高位为 0）。在有符号类型的情况下，最高位为 1 的字符转换为整数就是一个负数，所以可以使用连续的两个负数来判断一个字符是不是汉字。具体代码请参考【代码 1-2】。

【代码 1-2】

```
#include <stdio.h>
#include <string.h>

int main() {
    int i, l, num;
    char s[10];
    scanf("%s", s);
    l=strlen(s);
    for(num=i=0; i<l; i++)
        if(s[i]<0)
            num++;
    printf("共有%d 个汉字\n", num/2);      //因为一个汉字占 2 个字节，所以要除以 2
    return 0;
}
输入数据为：ACM 俱乐部
输出结果为：共有 3 个汉字
```

2．C++中的基本数据类型

C++中包含了 C 语言中的所有基本数据类型，此外在 C++中基本数据类型还多了 bool 类型。bool 类型是用来表示真假的，它的值只有两个，即 true 和 false。在 C 语言中，可以用整数类型来表示真假，非零值表示真而零则表示假。除此之外，C++还增加了一个关键字 const，使用 const 定义变量时必须同时初始化这个变量，表示这个变量的值在初始化后不可改变。至于在 C++中经常使用的 string 类型，其实不是基本数据类型，而是一个类。

1.2.2 构造数据类型

构造数据类型是根据已定义的一个或多个数据类型用构造的方法来定义的。也就是说，一个构造类型的值可以分解成若干个“成员”或“元素”。每个“成员”都是一个基本数据类型或又是一个构造类型。

在 C 语言中，构造数据类型有数组类型、结构体类型、枚举类型以及共用体（联合体）类型。在 C++中构造数据类型还包含类类型。在 C++中类类型与结构体类型基本是相同的，唯一的区别就是类类型在未声明限定符（public、private 等）时，类默认的是 private，而结构体默认为 public。

在 C++中有些类是 C++标准库中的类型，这些类型只要包含相应的头文件就可以直接被调用。例如，前面刚刚提到的用来表示字符串的 string 类型。在 C 语言中使用 char 类型的数

组来表示一个字符串，但在C++中可以直接使用string来表示字符串，当然前提是需要添加头文件<string>。

1.2.3 指针类型

指针向来都是C/C++初学者们最头疼的一块内容。其实，指针是一种特殊的数据类型，其值用来表示某个变量在内存中的地址，形象点说就是指针指向了某个内存中的变量。对于不同类型的指针，其指向的数据的类型是不同的。指针的值可以使用unsigned long来查看与表示，因而使用整数是可以输出该值的，也就是代表内存地址的那个数字。但指针与整数仍然是不同的类型，切勿混淆。

提醒：在上机考试中，除非是必须用到指针，否则不要用指针来做题，以免造成不必要的错误！

1.2.4 引用类型

在C语言中，函数参数的传递都是按照值传递的。比如说，我们在主函数中调用一个函数并向其中传递参数变量a和b时，被调用的函数会为a和b创建副本，然后将a和b的值赋给刚才创建好的副本。如果这两个副本的值在被调用函数中改变的话，主函数中的a和b的值并不会发生改变。当然，我们可以修改那个被调用的函数，改为向它传递变量a和b的指针，这样在函数中改变指针所指向的内容就可以实现主函数中的变量a和b的值也发生改变。

然而，在C++中多了一个引用类型。即在定义变量时在变量前加上一个取地址符“&”。例如：

```
int a;
int &b=a, &c=a;                    //b 和 c 就是 a 的两个引用
```

引用变量实际上是一个变量的别称，一个变量可以有多个引用。也就是说，一个变量可以有多个别称。在定义引用变量时并没有为被引用变量创建一个副本，或者说并没有为其分配新的内存空间。引用变量与被引用变量实际上是同一个变量的不同名字而已，这样如果改变这个变量本身的值的话，它的所有引用变量的值也会一同改变，反之亦然。【代码1-3】可以很直观地看出引用的性质。

【代码1-3】

```
#include <stdio.h>

void Half(int num) {                    //使用值传递
    num/=2;
}

void Half_2(int & num) {                //使用引用传递
    num/=2;
}

int main() {
```

```
    int num1=6, num2=6;
    Half(num1);
    Half_2(num2);
    printf("%d %d\n", num1, num2);  //输出使用值传递与使用引用传递的结果
    return 0;
}
```

输出结果为：6 3

1.2.5 空类型

一个函数一般需要传入参数以及返回相应的值。如果某个函数不必传入参数，则应将传入的类型定义为 void 类型。同样，如果没有返回值，那么返回值的类型也需要定义为 void。如【代码 1-3】中的 Half 与 Half_2 不需要传递返回值，所以返回值的类型定义为 void。犹如【代码 1-4】中的 CountFunNum 函数，既无须形参，又无须返回值，因而两者均为 void。当然，前面提到的这些情况都可以省略 void 这个关键字。

而当 void 类型与指针结合时，则表示该指针能够指向任意类型的变量。当然，如果需要获取该指针所指向的变量的值，则应当将该指针转换为相应的类型。例如，5.1.5 小节中的【代码 5-6】，Cmp 函数的形参 pa 和 pb 都是 const void*类型，而真正使用的应该是 int*类型，因此，在使用这两个参数时，需要将参数类型从 const void*强制转换为 int*。

1.3 语句

下面将介绍 C/C++中的一些常用语句，例如变量与数组的定义、循环的使用等。

1.3.1 变量的定义与初始化

要想定义并使用变量就要先知道变量有哪几种，每种变量的特性是什么，然后再了解变量如何定义，以及变量的作用域是什么。下面将按照这个顺序来讲解变量的相关知识。

1. 全局变量与局部变量

变量在使用前必须声明或定义。而定义变量既可以放在函数内部，也可以放在函数外部。在函数内部定义的变量称为局部变量，而定义在函数外部的变量则称为全局变量。局部变量在定义时所分配的空间位于内存中的“栈”。“栈”的特点是存取的速度比较快，但所具备的存储空间相对较少。全局变量在定义时所分配的空间位于内存中的一个叫做“堆”的部分。“堆”的特点是存取速度比较慢，但所具备的存储空间却相对较大。因此，定义变量时，我们需要根据实际情况决定是在函数内部定义还是在函数外部定义。

全局变量在定义时具有默认初始值。也就是说，全局变量在定义时即使没有指定初始值，该全局变量也具备了默认的初始值，例如整型全局变量的默认初始值为 0，字符串的默认初始值为空字符串。但局部变量不具备默认初始值，也就是说，如果定义一个局部变量，在没有赋值过的情况下使用该变量，该变量的值是不确定的。

全局变量在其定义后的所有函数中均能被访问，也就是说，如果在某个函数中改变了全局变量的值，那么在其他函数中使用到的该全局变量的值也会发生变化。如果代码规模很大，可能有很多函数调用了全局变量，那么此时将造成混乱，因而在编写软件时不常用全局变量，