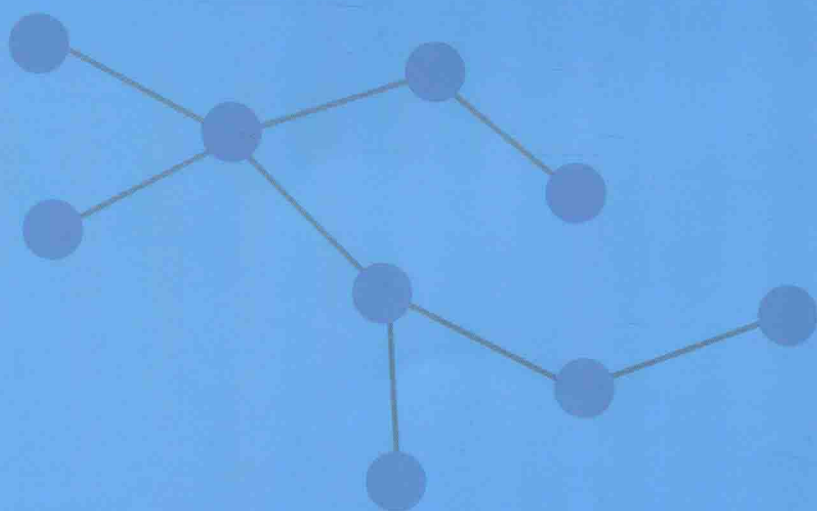


高等院校计算机任务驱动教改教材

C语言实验指导

—— 基于程序设计的在线评判系统

陈荣钦 主 编 王爱冬 副主编



清华大学出版社



高等院校计算机任务驱动教材

C语言实验指导

——基于程序设计的在线评判系统

陈荣钦 主 编 王爱冬 副主编

清华大学出版社
北 京

内 容 简 介

本书共分为三章,第1章主要介绍了实验环境,对几种常见的C语言开发工具及实验平台(TOJ)的使用进行了详细的介绍;第2章为基础应用,每个实验都在实验平台(TOJ)中精选了适合课程学习的题目,并有不同程度的提示帮助读者完成题目,非常适合于计算机及非计算机专业C语言的实验教学;第3章为高级应用,引入了程序设计竞赛中较为常见的基础算法,并配合相关的练习题。

本书适合本科及高职高专计算机相关专业学生阅读,也适合计算机初学者学习。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C语言实验指导:基于程序设计的在线评判系统/陈荣钦主编. --北京:清华大学出版社,2015
高等院校计算机任务驱动教改教材
ISBN 978-7-302-41022-5

I. ①C… II. ①陈… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第160477号



责任编辑:张龙卿

封面设计:徐日强

责任校对:袁芳

责任印制:宋林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印装者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:9.5

字 数:229千字

版 次:2015年10月第1版

印 次:2015年10月第1次印刷

印 数:1~1500

定 价:26.00元

产品编号:065676-01

前言

随着我国信息化进程的推进,各行各业对 IT 人才的需求量日益增加,计算机专业已成为我国第一大专业。然而,随着就业形势的日益严峻,计算机类专业本科毕业生就业难的问题开始凸显,一方面毕业生找不到满意的工作,另一方面用人单位感慨招不到适合需要的专业人才。调研表明,用人单位普遍反映计算机类应届毕业生存在以下几个较为突出的问题。

- (1) 应聘简历中课程繁多,但学生缺乏专业核心竞争力。
- (2) 缺乏解决实际问题的能力。
- (3) 缺乏创新意识和能力。
- (4) 自主学习能力偏差。
- (5) 普遍缺乏沟通和团队协作能力。

而其中程序设计能力是计算机专业学生最主要的核心竞争力之一,没有程序设计基础就无法更好地深入掌握计算机学科知识。学生程序设计能力的培养主要通过程序设计类课程实验教学、课外自主实践、项目实践等环节得以实现。而 C 语言作为程序设计类课程的第一门基础课,其教学效果的好坏往往直接关系到人才培养的成败,多数学生因没学好 C 语言课程而开始厌恶程序设计,因此影响了后续课程的学习,影响了专业能力的培养。

近几年 ACM/ICPC 国际大学生程序设计竞赛在我国相当火热,大部分程序爱好者对其充满了热情,很多参赛者也在就业和考研中证明了自己的实力,得到了单位的认可。同时不少高校也都建立了自己的在线评测(Online Judge,OJ)系统,2008 年以来,我们建立并不断完善了“台州学院在线程序设计综合实验平台(<http://acm.tzc.edu.cn>,简称 TOJ)”,并在 C 语言中探索如何使用 OJ 系统来进行实验教学,结果证明 OJ 系统在实验教学中的普及,极大地提高了学生的程序设计能力,我校学生也在省程序设计竞赛中连续取得了优异的成绩。

OJ 系统的使用在提高教学效果的同时,总体上却减轻了教师的教学负担。但我们也发现,由于缺乏相应的实验指导教材,学生在入门学习阶段遇到的问题较多,一部分学生在学习过程中需要教师更多地进行课外辅导,而市面上大部分的竞赛辅导教材并不适合于普通学生学习 C 语言。本书就是在这种情况下编写的。

本书共分为三章,第 1 章主要介绍了实验环境,对几种常见的 C 语言开发工具及 TOJ 的使用进行了详细的介绍。第 2 章为基础应用,每个实验都

在 TOJ 中精选了适合课程学习的题目,并有不同程度的提示帮助读者完成题目,非常适合于计算机及非计算机专业 C 语言的实验教学。第 3 章为高级应用,引入了程序设计竞赛中较为常见的基础算法,并配合了 TOJ 相关的练习题。每个题目都使用“TOJ+题号+标题”的形式给出,读者可以通过题号或者标题在 TOJ 中搜索到相应题目。

本书是在台州学院 ACM 集训队活动的基础上长期积累而成的,对台州学院集训队队员们以及在 TOJ 上长期做题的选手表示由衷的感谢。

由于水平有限,书中难免存在表达不当甚至错误之处,恳请读者给本书多提宝贵意见。编者 E-mail:chen_rongqin@163.com。

编 者

2015 年 4 月

目 录

第 1 章 实验环境介绍	1
1.1 程序设计实践平台	1
1.1.1 平台简介	1
1.1.2 在线实验教学的基本流程	2
1.1.3 TOJ 测试数据处理	2
1.1.4 常见错误及处理方法	7
1.1.5 C 语言编程风格	8
1.2 Microsoft Visual Studio 集成开发环境介绍	9
1.2.1 使用 Visual C++ 6.0 创建应用程序的基本步骤	9
1.2.2 使用 Visual C++ 6.0 调试应用程序	12
1.2.3 使用 Visual Studio 2012 创建应用程序的基本步骤	14
1.3 CFree 集成开发环境介绍	16
1.3.1 使用 CFree 创建应用程序的基本步骤	16
1.3.2 使用 CFree 调试应用程序	18
1.4 Code::Blocks 集成开发环境介绍	21
1.4.1 使用 Code::Blocks 创建应用程序的基本步骤	21
1.4.2 使用 Code::Blocks 调试应用程序	24
第 2 章 基础应用	28
2.1 实验 1 简单程序设计	28
2.1.1 实验目的	28
2.1.2 实验预习	28
2.1.3 实验任务	28
2.1.4 相关题库	33
2.2 实验 2 选择结构程序设计	35
2.2.1 实验目的	35
2.2.2 实验预习	36
2.2.3 实验任务	36
2.2.4 相关题库	41
2.3 实验 3 循环结构程序设计	45
2.3.1 实验目的	45

2.3.2	实验预习	45
2.3.3	实验任务	45
2.3.4	相关题库	52
2.4	实验 4 数组的使用	63
2.4.1	实验目的	63
2.4.2	实验预习	63
2.4.3	实验任务	63
2.4.4	相关题库	70
2.5	实验 5 函数的定义与调用	75
2.5.1	实验目的	75
2.5.2	实验预习	75
2.5.3	实验任务	75
2.6	实验 6 预处理命令	83
2.6.1	实验目的	83
2.6.2	实验预习	83
2.6.3	实验任务	83
2.7	实验 7 指针的使用	87
2.7.1	实验目的	87
2.7.2	实验预习	88
2.7.3	实验任务	88
2.8	实验 8 结构体的使用	96
2.8.1	实验目的	96
2.8.2	实验预习	97
2.8.3	实验任务	97
2.9	实验 9 文件	100
2.9.1	实验目的	100
2.9.2	实验预习	100
2.9.3	实验任务	100
2.10	实验 10 位运算	104
2.10.1	实验目的	104
2.10.2	实验预习	104
2.10.3	实验任务	104
第 3 章	高级应用	107
3.1	实验 1 筛选法求素数表	107
3.2	实验 2 高精度加减运算	109
3.2.1	高精度加法运算	109
3.2.2	高精度减法运算	111
3.3	实验 3 二分查找	113

3.4	实验4 快速排序算法	115
3.5	实验5 向量法求解多边形面积	118
3.6	实验6 向量法判断线段是否相交	121
3.7	实验7 简单贪心算法求解背包问题	124
3.8	实验8 简单动态规划求解0~1 背包问题	126
3.9	实验9 搜索算法求解迷宫问题	132
3.9.1	深度优先搜索	132
3.9.2	广度优先搜索	134
3.10	实验10 字典树	136
3.11	算法入门习题	139
附录一	Visual C++ 6.0 常见编译或链接错误信息	142
附录二	CFree 5.0 常见编译或链接错误信息	144

第 1 章 实验环境介绍

1.1 程序设计实践平台

1.1.1 平台简介

学习 C 语言最佳的方式是动手实践,一般遵照“分析问题,设计算法,编写代码,反复调试”的步骤学习,在线程序设计实践平台正是提供了这样的学习环境。本书结合台州学院在线程序设计综合实验平台(<http://acm.tzc.edu.cn>,以下简称 TOJ)编写,相关的实验教学过程均可在实验平台进行。平台目前拥有大量适合于 C 语言学习的中英文题目,并有难度估计、题目分类,有利于初学者循序渐进学习。系统主要功能如下。

(1) 用户管理:提供新用户注册及更新、加入班级和团队、用户排名和班级排名等功能。

(2) 题目管理:允许用户在线提交答案、永久保存并查看源代码、上传题目等。

(3) 在线比赛:自由参加在线竞赛并根据排名获取积分、自己组织比赛、撰写解题报告等。

(4) 实验管理:提供实验教学环境,在线完成实验题目、撰写实验报告、教师批改和打印、自动统计实验成绩和平时成绩等。

(5) 在线交流:提供站内邮件、Web 交流群、讨论版、论坛和公告等内容进行在线交流。

(6) 通关游戏:通过通关方式,让学生在游戏中的体会解题的快乐。

(7) 趣味活动:使用积分参与礼品兑换、图书借阅、竞赛竞猜等趣味活动。

用户在系统中解题的基本流程如图 1-1 所示。

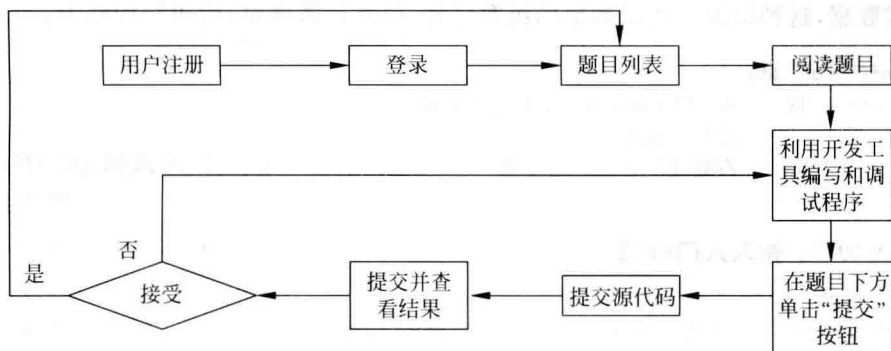


图 1-1 用户解题基本流程

1.1.2 在线实验教学的基本流程

实验平台提供的实验教学模块能基本满足“C 语言程序设计”课程的实验教学,可为教师提供更多班级管理功能,通过排名更容易了解学生的学习情况。基本流程如图 1-2 所示。

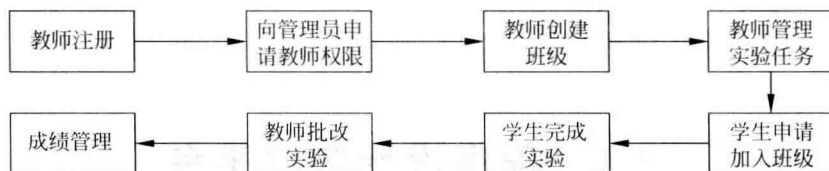


图 1-2 实验教学模块基本流程

- (1) 教师注册及权限申请: 教师先注册自己的账号,并联系管理员申请权限。
- (2) 创建班级: 通过个人菜单进入教师管理主页创建班级,填写基本信息。
- (3) 发布实验: 进入相应班级中的实验模块,发布实验任务,并添加相关实验题目。
- (4) 添加学生: 教师无权自行添加,需告知学生在班级页面中加盟,并通过教师后台审核。
- (5) 学生实验: 学生通过个人菜单进入“实验”模块(首次审核可能需要重新登录),完成相关题目,并填写实验报告。
- (6) 批改实验: 教师可以查看学生题目完成情况,并批改实验报告。
- (7) 成绩管理: 教师可以管理学生的平时成绩、实验成绩等,并可以打印输出。另外,教师可以在教学中创建班级比赛,并导出相关成绩。

1.1.3 TOJ 测试数据处理

实验平台的评判原理是通过后台大量测试数据来评测用户提交的程序,若程序能通过所有数据的测试,则认为程序是正确的。读题时需要注意题目输入/输出是单组还是多组,在处理多组数据时,用户只需要遵照“输入一组、处理一组、输出一组”的模式进行,无须一次性输入所有组数据。多组测试数据的输入主要分为以下几种情况。

(1) 已知测试数据组数

如 TOJ1077 题目所介绍,数据的第一行是一个整数 t ,表示测试数据的组数。接下来有 t 组测试数据,这种情况下测试数据的组数已知,只需计数即可,代码结构如下:

```

scanf("%d", &t);
while(t--){    //t 值不断减小,到 0 时结束循环
    //每组数据输入
    //处理后输出一组数据
}
  
```

【TOJ1077: 输入入门(1)】

题目描述

计算 $A+B$ 。

输入描述

输入第 1 行为一个整数 $n(1 \leq n \leq 10)$,代表测试的组数。下面有 n 组测试数据,每组

一行,为两个整数 A 、 B 。

输出描述

输出 $A+B$ 的值。

样例输入	样例输出
2	3
1 2	7
3 4	



参考程序

```
#include <stdio.h>

int main()
{
    int a, b, t;
    scanf("%d", &t);
    while(t-- )
    {
        scanf("%d%d", &a, &b);
        printf("%d\n", a+b);
    }
    return 0;
}
```

(2) 有结束标志

一般题目中说以某某数据结束都属于此类。如 TOJ1078 描述为输入数据有多组,以 0 0 结束,这种情况下测试数据组数未知,但在数据文件的结尾有一个结束标志,代码结构如下:

```
while(scanf("%d%d", &a, &b), a || b)//a || b 即  $a \neq 0 \vee b \neq 0$ , 则  $a$  或  $b$  中有一个为非 0
{
    //每组数据中还可能会有其他数据输入
    //处理后输出一组数据
}
```

【TOJ1078: 输入入门(2)】

题目描述

计算 $A+B$ 。

输入描述

输入数据有多组。每组一行,为两个整数 A 、 B 。输入以 0 0 结束。

输出描述

输出 $A+B$ 的值。

样例输入	样例输出
1 2 0 0	3



参考程序

```
#include <stdio.h>
int main(){
    int a, b;
    while(scanf("%d%d", &a, &b), a||b)
        printf("%d\n", a+b);
    return 0;
}
```

(3) 以 EOF 结束

输入数据有多组,以 EOF 结束,或者未告知数据组数,也没有结束标志的都属于此类。这种情况下,测试数据的组数是未知的,EOF(End of File)表示处理到文件结尾为止(测试数据在服务器以文件形式存在)。代码结构如下:

```
while(scanf("%d%d", &a, &b) != EOF)
{
    //每组数据中还可能会有其他数据输入
    //处理后输出一组数据
}
```

EOF 在这里实际上定义为 -1,用来判断 scanf 函数的返回结果。当 scanf 函数正常读取数据时,返回的值是读取的数据个数,如上例中正常应该返回 2。当读取到文件结束时,scanf 函数将返回值 -1。因此也可以写成以下形式:

```
while(scanf("%d%d", &a, &b) == 2)
{
    //每组数据中还可能会有其他数据输入
    //处理后输出一组数据
}
```

当学完“数组”一章后,可能会碰到不断读入字符串并以 EOF 结束的情况。如果字符串中不含空格,则形式与上述代码一致;但若字符串中含有空格(如 TOJ4551),则可能需要使用 gets 读取整行(scanf 碰到空格就结束 1 个数据的输入),此时的处理形式为:

```
while(gets(str) != NULL)    //NULL 即为 0,因此可以修改为 while(gets(str)),str 为字符数组名
{
    //每组数据的处理及输出
}
```

【TOJ1076: 输入入门(3)】

题目描述

计算 $A+B$ 。

输入描述

输入数据有多组。每组一行,为整数 A 、 B 。以 EOF 做结束。

输出描述

输出 $A+B$ 的值。

样例输入	样例输出
1 2	3
3 4	7



参考程序

```
#include <stdio.h>

int main(){
    int a, b;
    while(scanf("%d %d", &a, &b) != EOF)
        printf("%d\n", a+b);
    return 0;
}
```

【TOJ4551: 输入入门(4)】

题目描述

求字符串的长度。

输入描述

输入数据有多组,每行包含一个字符串,字符串可能包含空格字符。

输入直到文件结尾位置(即以 EOF 结束)。

输出描述

每组输出一个整数,即字符串的长度(包含空格字符在内)。

样例输入	样例输出
ab	2
a b	3
a b c	5



参考程序

```
#include <stdio.h>
#include <string.h>
int main(){
    char str[1001];
    while(gets(str) != NULL)
        printf("%d\n", strlen(str));
    return 0;
}
```

在处理多组数据的输出时,有时候需要正确地处理输出格式(如 TOJ1079),常见的输出格式有以下几种。

(1) 每组数据之间没有空行: 这种情况下不需要特别处理。

(2) 每组数据之后有一个空行: 这种情况下需要在每组之后输出换行符“printf("\n")”。

(3) 每组数据之间保留一个空行: 这种情况下空行的数目比实际的数据组数要少 1 个,因此需要特别处理。基本思路是除了第一组数据之外,每组数据输出之前先输出 1 个空行。代码结构如下:

```
flag = 0;           //在没有任何输入时标记为 0
while(输入并检测结束条件)
{
    if(flag)        //第一组由于标记为 0,因此不输出空行,后面已经标记为 1,均要输出换行
        printf("\n");
    flag = 1;       //输入一组后标记为 1
    //输出一组数据
}
```

【TOJ1079: 输出入门】

题目描述

计算 $A+B$ 。

输入描述

输入数据有多组。每组一行,为两个整数 A 、 B 。输入以 0 0 结束。

输出描述

输出 $A+B$ 的值,每组数据之间保留一个空行。

样例输入	样例输出
1 2	3
3 4	
0 0	7



参考程序

```
#include <stdio.h>
int main()
{
    int a, b, flag = 0;
    while(scanf("%d%d", &a, &b), a||b){
        if(flag)
            printf("\n");
        flag = 1;
        printf("%d\n", a + b);
    }
    return 0;
}
```

1.1.4 常见错误及处理方法

实践平台中的评判结果若为 Accepted, 表示程序已经完全通过了系统的测试数据, 程序正确。除此之外, 系统还给出了其他一些结果, 其中以下几个不属于代码错误。

(1) Waiting(等待): 系统正在做评判的准备工作, 需稍等片刻再刷新浏览器查看结果。

(2) Judging(评判中): 系统正在评判程序, 需稍等片刻再刷新浏览器(一般为 F5 快捷键)查看结果。

(3) System Error(系统错误): 系统内部出现错误, 需及时与管理员联系。

以下结果一般为代码错误。

(1) Compile Error(编译错误): 程序语法有问题, 编译器无法编译, 平台使用的编译器是 GNU GCC, 在其他编译器上正常编译的程序并不能完全保证在 GCC 上正常编译。具体的出错信息可以单击“Compile Error 链接”查看, 并根据提示信息修改错误。

(2) Output Limit Exceeded(输出超过限制): 程序向控制台输出了太多信息, 一般是因为程序存在死循环, 而在死循环中存在输出语句, 因此输出会超过限制。如 while(scanf("%d", &a)) 这种语句的出现, 即使到了文件结尾也不会停止输入。

(3) Time Limit Exceeded(超时): 程序运行的时间已经超出了这个题目的时间限制, 可能算法效率太低, 或者出现死循环耗时太多。

(4) Presentation Error(格式错误): 程序的输出格式存在问题, 需要检查程序的输出是否多了或者少了空格(' ')、制表符('\t')或者换行符('\n')。

(5) Wrong Answer(答案错误): 程序已经正常运行并输出了结果, 但答案错误, 要注意每个题目的后台数据一般比较多, 通过了样例中的数据并不等于程序正确, 用户需要输入更多的数据(尤其是一些边界数据), 并对程序进行仔细的测试。

(6) Runtime Error(运行时错误): 运行时错误, 一般是程序在运行期间执行了非法的操作造成的, 表明程序在运行后台数据的过程中出现错误被非法中断。以下列出常见的错误类型。

- Runtime Error (ARRAY_BOUNDS_EXCEEDED) // 数组越界。
- Runtime Error (DIVIDE_BY_ZERO) // 除零, 分母可能为 0, 调试求余、相除的代码。

- Runtime Error(Access_Violation)//非法内存访问,可能使用了未分配的指针,或者数组下标错误而访问非法的内存空间。
- Runtime Error(Stack_Overflow)//栈溢出,栈的空间一般只有 2MB,可能定义的局部数组过大,或者递归函数调用太深。

(7) Memory Limit Exceeded(内存超过限制): 程序运行的内存已经超出了这个题目的内存限制,可能分配的数组太大。

在碰到错误时,读者一般应该检查以下几个方面的问题。

(1) 阅读题目的输入/输出描述,仔细检查变量的类型是否完全一致,比如输入为小数的绝不能用整型变量输入;输入/输出中是否有其他特定的符号,字母的大小写是否有问题,有没有多余或缺少空行、空格,整数是否超过 32 位(二进制)等。

(2) 阅读题目的输入/输出描述,测试题目中的边界数据是否有问题,比如题目有要求 $1 \leq N \leq 100000$ 之类的语句,应当测试 N 为 1 和 100000 时程序是否正确,数组定义是否够大等。

(3) 若碰到超时,仔细检查程序的效率,即最大的循环次数,一般情况下循环次数在百万、千万级别时,就应该要考虑是否是因为效率问题;否则应该考虑是否是死循环造成的,需要仔细检查循环条件(比如循环条件写错,循环内部修改了控制变量等)。

当整数超过 32 位(二进制)时,即某变量的值超过 $2^{31} - 1 = 2147483647$,如果该变量值不超过 $2^{63} - 1 = 9223372036854775807$,此时可以使用 64 位整型(二进制):

(1) Windows 平台下为 long long 或 __int64(两个下划线),输出格式为 %I64d, TOJ 为该方式。

(2) Linux 平台下为 long long,输出格式为 %lld。

若整数值超过 64 位整型(二进制),此时应当使用数组形式来存储一个数,即使用数组的一个元素来存储一个数字,此时整数的各种运算都不能直接作用于数组上,需要逐位模拟。

1.1.5 C 语言编程风格

C 语言程序的编写风格虽然非常自由,但保持良好的编程风格对于团队开发和系统维护都有很大的帮助,因此遵循某种编程风格非常重要,以下是常见的基本编程风格。

(1) 代码缩进。在函数、选择结构、循环结构中,下一级的代码需要缩进,使程序结构清晰。在缩进时使用 Tab 键,而不是使用多个空格。

良好的风格	不好的风格
<pre>int main(){ int a, b, t; scanf("%d", &t); while(t--){ scanf("%d%d", &a, &b); printf("%d\n", a+b); } return 0; }</pre>	<pre>int main(){ int a, b, t; scanf("%d", &t); while(t--){ scanf("%d%d", &a, &b); printf("%d\n", a+b); } return 0; }</pre>

- (2) 短的语句也占一行,这样会让程序可读性更佳,并便于调试。
- (3) 大括号“{”和“}”也应尽量各占一行。
- (4) 变量名、函数名等标识符的命名应当做到见名知意,而且大小写风格应当一致。宏名、常量名一般全部使用大写,变量名一般全部使用小写,函数名中每个单词的首字符一般使用大写,其他字符使用小写。
- (5) 函数应当尽可能短小清晰,功能单一。
- (6) 二元运算符的两侧往往应增加空格,使表达式更加清晰;如 $a=b$ 。
- (7) 函数定义、复杂的语句或语句块应当有适当的注释语句,用以解释程序的功能。
- (8) 表达式尽量简单,如果过于复杂,可以分成多个语句。
- (9) 尽量使用括号来决定优先级,而不是使用默认的优先级顺序。
- (10) 除非必需,尽量少用全局变量和静态局部变量。
- (11) 除非必需,尽量不用 goto 语句。
- (12) 编程风格不是绝对的,应该与整个团队的风格保持一致。

1.2 Microsoft Visual Studio 集成开发环境介绍

本节主要介绍如何通过 Visual C++ 6.0 创建、编写、编译、运行和调试一个简单的 C 语言程序,并简要阐述 Visual Studio 2012 创建应用程序的过程。

1.2.1 使用 Visual C++ 6.0 创建应用程序的基本步骤

1. 启动 Visual C++ 6.0

单击“开始”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令,启动 Visual C++ 6.0 应用程序,其启动界面如图 1-3 所示。

2. 创建项目

Visual C++ 6.0 中的项目用来管理多个源文件,创建项目的方法是单击 File(文件)→New(新建)命令,在弹出的 New(新建)对话框中(见图 1-4)选择 Projects(项目)选项卡,并在列表中指定项目类型为 Win32 Console Application,在右侧的 Project name 项自行填写项目名称,在 Location 一栏单击“...”浏览按钮选择项目存放的路径(如本例中路径为 E:\demo1),然后单击 OK(确定)按钮,在后续弹出的对话框中依次单击 Finish(完成)按钮和 OK(确定)按钮完成一个空项目的创建。若要查看创建的项目文件,可进入项目所保存的文件夹中,找到 demo1.dsw 文件和 demo1.dsp 文件,其中扩展名为 .dsw 的文件为 workspace(工作区间)文件,用于管理多个 dsp 文件,一个 dsp 文件表示一个 project(项目),它将管理后续步骤中加入的 .cpp 或 .c 源文件等,只要使用 Visual C++ 6.0 打开 dsw 文件,即可打开工作区间中创建的所有项目(注意:仅仅打开 .c 或 .cpp 文件,则无法打开对应的项目)。

3. 创建源程序文件

再次单击 File(文件)→New(新建)命令,打开 New(新建)对话框(见图 1-5)。此时选择 Files(文件)选项卡,并在文件类型列表中选择 C++ Source File,在右侧中,保证已经选中