

总主编 刘 群
分主编 张 红
副主编 王 磊
编 委 陈 旭
编 委 陈 旭

青少年信息学 奥林匹克竞赛 实战辅导丛书

程序设计 与应用

(中学·C/C++)



东南大学出版社

青少年信息学奥林匹克竞赛实战辅导丛书

程序设计与应用

东南大学出版社

· 南 京 ·



内 容 提 要

程序设计涉及语言、环境和应用三个方面,学习程序设计的关键是培养融合这三个方面的系统化思维方法。针对起步阶段的中小学学生,如何利用短暂的课外时间,在短时间内达到较好的效果,是值得思考的问题。本书按照认知的规律,第1~第3章首先介绍计算机基本知识以及利用其工作需要的语言和算法的相关概念;第4~第10章以ANSI C和Dev-C++语言为例介绍计算机语言的相关知识及其基本使用;在此基础上,第11~第12章面向应用,介绍基本的数据结构、基础算法及其应用;附录部分详细总结C/C++语言的知识以及其开发环境的使用和调试技巧。考虑到实战训练特点,本书精选上百个信息学竞赛试题作为案例,采用问题驱动方式进行讲解,将知识点融入实例,注重程序设计技巧的解析,从而,实现程序设计系统化思维方法的训练。

本书主要满足广大中小学学生学习程序设计的教学和训练需求。同时,本书也非常适合作为普通高等学校本科以及专科学生学习程序设计课程的教学和学习参考用书。对一般的程序设计爱好者,本书也具有重要的参考价值。

图书在版编目(CIP)数据

程序设计与应用. 中学·C/C++ / 曹文主编. —南京:
东南大学出版社, 2010. 1

(青少年信息学奥林匹克竞赛实战辅导丛书)

ISBN 978-7-5641-2038-2

I. 程… II. 曹… III. C语言—程序设计—青少年读物
IV. TP311.1-49

中国版本图书馆 CIP 数据核字(2010)第 012715 号

东南大学出版社出版发行
(南京四牌楼2号 邮编 210096)

出版人:江 汉

江苏省新华书店经销 兴化市印刷有限责任公司印刷

开本:787 mm×1092 mm 1/16 印张:18 字数:544千字
2010年1月第1版 2010年1月第1次印刷 印数:1—4000册
定价:30.00元

(凡因印装质量问题,可直接向发行科调换。电话:025-83795801,网址:press.seu.edu.cn)

丛书序

得益于计算机工具的特殊结构,以计算机技术为核心的信息技术现在已在整个社会发展中起到了其重要的作用。同时,由于信息技术的本质在于不断创新,因而人们将 21 世纪称之为信息世纪。根据人类生理特征,青少年时期正处于思维活跃、充满各种幻想的黄金年代,孕育着创新的种子和潜能。长期的实践活动告诉我们,青少年信息学奥林匹克竞赛可以让广大的青少年淋漓尽致地展现其思维的火花,享受创新带来的美感。因此,该项活动得到了全国各地广大青少年朋友的喜爱,越来越多的青少年朋友怀着浓厚的兴趣加入到这项活动中来。

从本质上看,计算机学科是一种思维学科,正确地思维训练可以播种持续创新的优良种子。相对于其他学科的竞赛,信息学竞赛覆盖知识面更为宽广,涉及了数学、数据结构、算法、计算几何、人工智能等相关的专业知识。如何在短时间内有效地掌握这些知识的主体,并能灵活地应用其解决实际问题,显然是一个值得认真思考的问题。

知识学习与知识应用基于两种不同的思维策略,尽管这两种策略的统一本质上依赖于选手自身的领悟,但是如何建立两种策略之间的桥梁、快速地促进选手自身的领悟,显然是教材以及由其延伸的教学设计与实施过程所应考虑的因素。竞赛训练有别于常规的教学,要在一定的时间内得到良好的效果,需要有一定的技术方法,而不应拘泥于规范。从学习的本质看,各种显性知识的学习是相对容易的。或者说,只要时间允许,总是可以消化和理解的。然而,隐性知识的学习和掌握却是较难的。由于隐性知识的学习对竞赛和能力的提高起到决定性的作用,因此,仅仅依靠选手自身的感悟,而不能从隐性知识的层面重新组织知识体系,有目的地辅助选手自身的主动建构,显然是不能提高竞赛能力的。基于上述认识,结合多年来开展青少年信息学竞赛活动的经验,我们组织了一批有长期一线教学经验的教练员和专家、教授,编写出版了这套《青少年信息学奥林匹克竞赛实战辅导丛书》。

丛书的主要特点如下:

1. 兼顾广大青少年课外学习时间的短暂与知识内容较多的矛盾,考虑我国青少年信息学竞赛的特点和安排,丛书分为四个层次,分别面向日常常规训练、数据结构与数学知识强化、重点专题解析和精选试题解析,



既考虑知识体系的系统性及连续训练的特点,又考虑各个层次选手独立训练的需要。

2. 区别于常规的教学模式,丛书中每册书的体系设计以实战需要为核心主线,突出重点,整个体系从逻辑上构成符合某种知识体系学习规律的系统化结构。

3. 围绕实战辅导需求,在解析知识和知识应用关系所蕴涵的递归思维策略的基础上,重构知识点关系,采用抛锚式和支架式并重教学思路,突出并强化知识和知识应用两者之间的联系。

4. 在显性知识及其关系基础上,强调知识应用模式及其建构的学习方法的教学,注重学习思维和能力的训练,实现知识应用能力和竞赛能力的提高,强化从程序设计及应用的角度来进行训练的特点。

5. 整套丛书的设计,不仅注重竞赛实战的需要,还考虑选手未来的发展,强调计算机程序设计正确思维的训练和培养,以不断建立持续创新的源泉。

值此邓小平同志“计算机的普及要从娃娃抓起”重要讲话发表 25 周年之际,我们期望以此奉献给广大读者朋友一套立意新、选材精、内容丰富的青少年信息学奥赛读本。

本套丛书的编写与出版,得到了东南大学出版社的大力支持,在此表示感谢!

李立新 沈 军 王晓敏

2008 年 12 月



前言

在中国计算机学会的组织下,江苏省青少年科技中心已连续多年成功举办全国信息学奥林匹克联赛(简称 NOIP)活动,数以十万计的青少年从中受益。在这么多年的联赛活动中,参与此项工作的老师与专家积累了许多宝贵经验,从 1999 年起陆续撰写出版了两套青少年信息学奥林匹克竞赛丛书,包括初级本、中级本、高级本及全国青少年信息学奥林匹克联赛试题解析等。

现根据活动普及与发展的需要及广大读者的强烈建议,我们将这套丛书修订后呈现给大家。其中,本书以信息学奥林匹克竞赛初学者为主要对象,以“程序设计技巧”为主线,重在培养学生解决实际问题的能力,主编曹文具有 20 年信息学竞赛辅导经验。本书精选上百个信息学竞赛试题,深入浅出地介绍了语法和常用算法,对提高参赛选手的综合能力将起极为重要的引导作用。参加本书编写工作的有曹文、张信和秦新华,其中第 2、第 4、第 5、第 6 章和附录 A 由秦新华编写,第 3、第 7、第 8、第 9 章和附录 C 由张信编写,第 1、第 10、第 11、第 12 章和附录 B 由曹文编写。全书由东南大学计算机学院沈军教授统一审稿。

希望广大读者提出宝贵意见和建议,以便进一步修改,使之日趋完善。

编者

2009 年 12 月 20 日



目 录



第1章 C语言概论

1.1 C语言的发展过程	1
1.2 C语言的特点	1
1.3 C源程序的结构特点	2



第2章 认识计算机

2.1 进制转换	5
2.1.1 计算机是智能化的电器设备	5
2.1.2 二进制数的运算法则	5
2.1.3 十进制与二进制、八进制、十六进制之间的相互转换	5
2.2 计算机硬件知识	6
2.3 计算机工作原理	7
2.4 计算机软件知识	8
2.5 程序和算法	9



第3章 算法及算法的描述

3.1 算法的概念	10
3.2 算法的描述方法	10
3.2.1 自然语言描述	11
3.2.2 流程图描述	11
3.2.3 N-S图描述	14
3.2.4 伪代码描述	15
3.3 算法分析	16
3.3.1 时间复杂度	16
3.3.2 空间复杂度	17
3.3.3 时空的转换	17



3.3.4 算法分析及优化举例	18
3.4 C语言程序的基本结构	20
3.4.1 预处理指令	20
3.4.2 main 函数	21
3.4.3 保留字	22
3.4.4 标准标识符	22
3.4.5 用户标识符	22
3.4.6 大写与小写字母	23
3.4.7 注释	23
3.4.8 程序风格	24
本章小结	25



第4章 数据类型、运算符与表达式

4.1 数据类型与大小	26
4.2 常量与变量	28
4.2.1 常量和符号常量	28
4.2.2 变量	29
4.3 算术运算符	30
4.3.1 算术运算符简介	30
4.3.2 算术表达式	31
4.4 类型转换	31
4.5 自增与自减运算符	32
4.6 按位运算符	33
4.7 变量赋值	35
4.7.1 赋值符号与赋值表达式	35
4.7.2 变量赋值的要素	36
4.8 常用数学函数	37
4.8.1 三角函数	37
4.8.2 其他函数	38
4.8.3 幂指数	40



4.8.4 双曲函数	42
本章小结	43



第5章 数据输入输出的概念及在C语言中的实现

5.1 printf 函数(格式输出函数)	44
5.1.1 printf 函数调用的一般形式	44
5.1.2 格式字符串	45
5.2 scanf 函数(格式输入函数)	47
5.2.1 scanf 函数的一般形式	47
5.2.2 格式字符串	48
5.3 字符数据的输入输出	50
5.3.1 putchar 函数(字符输出函数)	50
5.3.2 getchar 函数(字符输入函数)	51
本章小结	53



第6章 选择结构程序设计

6.1 关系运算符和关系表达式	54
6.1.1 关系运算符及其优先次序	54
6.1.2 关系表达式	54
6.2 逻辑运算符和逻辑表达式	55
6.2.1 逻辑运算符及其优先次序	55
6.2.2 逻辑表达式	55
6.2.3 条件运算符	56
6.3 if 语句	57
6.3.1 if 语句	57
6.3.2 if 语句的嵌套	60
6.4 switch 语句	61
6.5 程序举例	63



6.6 语句与程序块	64
本章小结	65



第7章 循环控制

7.1 循环结构介绍	67
7.1.1 while 语句	67
7.1.2 for 语句	69
7.1.3 do while 语句	72
7.2 循环的嵌套	72
7.3 循环中断语句	74
7.3.1 break 语句	75
7.3.2 continue 语句	76
7.3.3 return 语句和 exit 语句	78
7.4 程序举例	78
7.5 常见错误	82
本章小结	84
习题7	85



第8章 数据组织与处理

8.1 数据组织	87
8.2 数据的处理与表达	91
8.2.1 声明数组	91
8.2.2 引用数组	91
8.2.3 数组初始化	94
8.2.4 数组元素的插入与删除	96
8.3 应用举例	97
8.3.1 简单应用	97
8.3.2 查找和排序	99



8.4 常见错误	110
本章小结	111
习题 8	112



第9章 函 数

9.1 函 数	114
9.1.1 函数原型	115
9.1.2 函数定义	116
9.1.3 函数调用	117
9.1.4 函数返回值	117
9.1.5 数组参数	119
9.1.6 标识符的作用域	120
9.2 递归及其实现	122
9.2.1 递归的概念	122
9.2.2 编写递归函数	123
9.2.3 递归函数的执行过程	124
9.2.4 递归的效率和优化	126
9.3 递归算法举例	127
9.4 结构化程序设计思想	135
9.4.1 结构化程序设计思想	135
9.4.2 自顶向下、逐步细化	135
9.4.3 单入口、单出口和三种基本结构	137
9.5 常见错误	139
本章小结	140
习题 9	141



第10章 指 针

10.1 指针与地址	143
10.1.1 指针的概念	143



10.1.2	指针变量的类型说明	144
10.1.3	指针变量的赋值	145
10.1.4	动态分配内存	145
10.2	指针与数组	147
10.2.1	指针与数组的关系	147
10.2.2	指向数组的指针	147
10.2.3	地址算术运算	149
10.2.4	编程实例	150
10.3	字符串及其处理	151
10.3.1	字符串的概念	151
10.3.2	字符串常用函数与操作	151
10.3.3	字符串应用举例	154
10.4	指针与函数	155
10.4.1	指针作为函数参数	155
10.4.2	函数返回指针	157
习题 10		158



第 11 章 基本数据结构及应用

11.1	结 构	159
11.1.1	结构的定义	159
11.1.2	结构变量的赋值	161
11.1.3	结构数组	162
11.1.4	结构指针	162
11.1.5	自引用结构	165
11.2	链 表	166
11.2.1	链表的概念	166
11.2.2	链表的建立、插入与删除	167
11.2.3	双向链表与循环链表	171
11.3	栈和队列	174
11.4	二叉树	179



11.4.1	基本概念	179
11.4.2	二叉树的建立	182
11.4.3	二叉树的遍历	182
11.4.4	排序二叉树	185
11.4.5	堆	187
习题 11		190



第 12 章 常用算法介绍

12.1	穷举法	192
12.2	回溯法	195
12.3	贪心法	201
12.4	分治法	208
12.5	递推法	214
12.6	动态规划	218



附录

附录 A	库函数	225
A.1	标准输入输出库: <stdio. h>	225
A.1.1	文件操作	225
A.1.2	格式化输出	227
A.1.3	格式化输入	229
A.1.4	字符输入输出函数	231
A.1.5	直接输入输出函数	233
A.1.6	文件定位函数	233
A.1.7	错误处理函数	234
A.2	数学函数: <math. h>	234
A.3	字符类测试函数: <ctype. h>	237
A.4	字符串函数: <string. h>	238
A.5	实用函数: <stdlib. h>	241



附录 B C++常用库与相关函数简介	245
B.1 流	245
B.2 字符串	248
B.3 二元集合(位集合)	251
B.4 算法	253
附录 C Dev-C++基本操作	255
C.1 Dev-C++简介及其安装	255
C.1.1 Dev-C++的安装	255
C.1.2 首次运行设置	257
C.2 集成环境简介	259
C.3 程序的输入及运行	262
C.4 程序的调试	264
C.4.1 设置断点	264
C.4.2 调试执行	265
C.4.3 使用查看	266
C.4.4 递归的调试	268
C.4.5 Dev-C++调试功能的不足	268
C.5 调试的相关知识	269
 参考文献	271



第 1 章 C 语言概论

1.1 C 语言的发展过程

C 语言是在 20 世纪 70 年代初问世的。1978 年美国电话电报公司(AT&T)贝尔实验室正式发布了 C 语言。同时由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的 *THE C PROGRAMMING LANGUAGE* 一书,通常简称为 K&R,也有人称之为 K&R 标准。但是,在 K&R 中并没有定义一个完整的标准 C 语言,后来由美国国家标准学会在此基础上制定了一个 C 语言标准,于 1983 年发表。通常称之为 ANSI C。

早期的 C 语言主要用于 UNIX 系统。随着 C 语言的强大功能和各方面的优点逐渐为人们认识,到了 20 世纪 80 年代,C 开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用。成为当代最优秀的程序设计语言之一。

1.2 C 语言的特点

1. 简洁紧凑、灵活方便

C 语言一共只有 32 个关键字、9 种控制语句,程序书写自由,主要用小写字母表示。C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来,可以像汇编语言一样对位、字节和地址进行操作,而这三者是计算机最基本的工作单元。

2. 运算符丰富

C 的运算符包含的范围很广泛,共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 的运算类型极其丰富、表达式类型多样化。灵





活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据结构丰富

C 的数据类型有:整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂的数据类型的运算;同时,引入了指针概念,使程序效率更高。另外 C 语言具有强大的图形功能,支持多种显示器和驱动器,计算功能、逻辑判断功能也十分强大。

4. 结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言主要由函数形式构成,这些函数可方便的调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

5. 语法限制不太严格、程序设计自由度大

一般的高级语言语法检查比较严格,能够检查出几乎所有的语法错误。而 C 语言给予程序编写者有较大的自由度。

6. 允许直接访问物理地址,可以直接对硬件进行操作

7. 程序生成代码质量高,程序执行效率高

C 语言一般只比汇编程序生成目标代码的效率低 10%~20%。

8. 适用范围大,可移植性好

C 语言的一个突出优点就是适用于多种操作系统,如 DOS、UNIX;也适用于多种机型。

1.3 C 源程序的结构特点

以下几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从中了解到一个 C 语言源程序的基本组成部分和书写格式。

```
egl
最简单的程序
int main() {
    return 0;
}
```

这是一个最基本的程序的框架,所有的 C 语言程序都是从此框架上构建出来的。





eg2

输出 hello world

```
#include<stdio. h>
int main() {
    printf("hello world! \n"); /* 输出 */
    return 0;
}
```

这个程序在基本框架上稍稍修改,调用了标准输入输出库,并用输出语句,输出了 hello world。

eg3

输入 x,输出函数 $f(x)=|x-10|$ 的值

```
#include<stdio. h>
int main() {
    printf("Input the value of x : ");
    int x;
    scanf("%d",&x);/* 读入自变量的值 */
    printf("f(x) = ");
    if(x-10<0) printf("%d\n",10-x); /* 判断自变量的取值在哪一个分段
内 */
    else printf("%d\n",x-10);
    return 0;
}
```

这里使用了语句 if,用于判断自变量所属的区间。

eg4

输入 n,输出一个用 * 组成的高为 n 的直角三角形

```
#include<stdio. h>
int main() {
    int n;
    printf("Input the Size of the triangle : ");
    scanf("%d",&n);/* 读入三角形大小 */
    int i;
    for(i=0;i<n;i++) {
```

