

新大纲

The National Computer Rank Examination

**全国计算机等级考试
(二级、C 语言程序设计)
上机考试试题分析
与应试指南**

■ 司存瑞 编著



西安电子科技大学出版社

<http://www.xduph.com>

全国计算机等级考试
(二级、C 语言程序设计)
上机考试试题分析与应试指南

司存瑞 编著

西安电子科技大学出版社

2001

内 容 简 介

本书精选了近年来全国计算机等级考试(二级、基础知识和 C 语言程序设计)上机考试中的典型试题,这些试题几乎覆盖了全国计算机等级考试(二级、基础知识和 C 语言程序设计)上机考试中的全部内容。本书不同于一般的习题集或者分类试题集,而是注重于试题分析。在分析的过程中尽可能全面、详细地介绍计算机基础知识和 C 程序设计中遇到的语法内容、使用方法和程序设计技巧。全书共分五章,内容包括 C 语言概述、Turbo C 2.0 程序设计环境与程序调试技术、全国计算机等级考试(二级、C 语言程序设计)上机考试应试指南、历年全国计算机等级考试上机试题分析和计算机等级考试上机考试模拟试题。

本书不但可供参加全国计算机等级考试的考生使用,同时也可供高等院校学生和各级各类人员学习计算机基础与 C 语言程序设计者使用。对于从事计算机方面教学和科研的教师、科技人员,本书也很有参考价值。

图书在版编目(CIP)数据

全国计算机等级考试(二级、C 语言程序设计)上机考试试题分析与应试指南

/司存瑞编著. —西安:西安电子科技大学出版社,2001.10

ISBN 7-5606-1071-4

I. 全… II. 司… III. 电子计算机—水平考试—自学参考资料 IV. TP3

中国版本图书馆 CIP 数据核字(2001)第 067639 号

责任编辑 霍小齐 张晓燕

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)8227828 邮 编 710071

http: //www. xduph. com E-mail: xdupfxb@pub. xaonline. com

经 销 新华书店

印 刷 西安文化彩印厂

版 次 2001 年 10 月第 1 版 2001 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 16.5

字 数 387 千字

印 数 1~6000 册

定 价 22.00 元

ISBN 7-5606-1071-4/TP·0532

* * * 如有印装问题可调换 * * *

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

前 言

随着计算机技术在我国各个领域的推广、普及,计算机作为一种广泛应用的工具,其重要性日益受到社会的重视。越来越多的人开始学习计算机,越来越多的单位和部门已把掌握一定的计算机知识和应用能力作为干部录用、职称评定、职务晋升、上岗资格的重要依据之一。为此,国家有关部门在全国范围陆续推出了一些计算机类考试,如全国计算机等级考试、中国计算机软件专业技术资格和水平考试、计算机及信息高新技术职业技能鉴定、全国高等教育自学考试、普通高校非计算机专业学生计算机应用知识能力等级考试等。这些考试为有关部门公正、客观地选择工作人员提供了参考,同时也促进了计算机知识的全面普及和应用。

纵观近几年的各类考试情况,参加全国计算机等级考试的人数最多。等级考试二级主要考核应试者对软、硬件基础知识的掌握和使用一种计算机程序设计语言(QBASIC、FORTRAN、PASCAL、C、FoxBASE)编制程序和上机调试的能力,而报考C语言程序设计的人数占有相当大的比例。这是因为C语言作为一门理想的结构化程序设计语言,具有功能丰富、使用灵活、应用面广、编写的程序效率高及可移植性好等优点。在全国计算机等级考试的三级和四级中,C语言是惟一选用的高级语言。因此,对这种考试的形式和内容进行分析和研究,无疑会对提高我们的计算机知识水平和应用能力起到一定的积极作用。为此,作者撰写了《全国计算机等级考试(二级)试题分析与应试指南——基础知识与C语言程序设计》一书。该书出版后,受到读者普遍好评;同时,不少读者也反映,该书偏重于笔试,对于上机考试,也需要类似的这样一本书。事实上,从近几年参加全国计算机等级考试(二级、C语言程序设计)的成绩统计分析来看,笔试部分平均通过率约为40%左右,而上机部分通过率约为30%左右,这也正是读者热切希望能有《全国计算机等级考试(二级、C语言程序设计)上机考试试题分析与应试指南》一书的真正原因。本书正是为此目的而撰写的。

本书收集了近年来全国计算机等级考试(二级、基础知识和C语言程序设计)上机考试中的典型试题,这些试题几乎覆盖了全国计算机等级考试(二级、基础知识和C语言程序设计)上机考试中的全部内容。本书不同于一般的习题集或者分类试题集,而是注重于试题分析。在分析的过程中尽可能全面、详细地介绍计算机基础知识和C程序设计中所遇到的语法内容、使用方法和程序设计技巧。全书共分五章,内容包括C语言概述、Turbo C 2.0程序设计环境与程序调试技术、全国计算机等级考试(二级、C语言程序设计)上机考试应试指南、历年全国计算机等级考试上机试题分析和计算机等级考试上机考试模拟试题。

作者力图把本书写得比较实用,但由于学习者在上机过程中可能遇到各种各样的问题,作者不可能在有限的篇幅内将初学者遇到的所有问题包含在内,作者也从未有过编写一本庞大的上机指南的设想。相反地,倒是希望学习者能够认真地阅读并掌握第一、二章的基本内容和程序调试方法,在此基础上,再来编写和调试第四、五章给出的为数不是很

多但却是十分精选的试题，并且能够举一反三，触类旁通。通过这种方法，相信读者编写程序与调试程序的水平和能力会有较大幅度的提高。如果这样，作者将会感到十分欣慰。

虽然此书原稿一直作为大专院校开设“C 语言程序设计”这门课程的教学内容在课堂和等级考试辅导班对学生多次讲授，并且取得了良好的效果，但由于作者水平有限，错误和缺点在所难免，可能会有许多不尽如人意之处，恳望读者不吝赐教。

本书在编写过程中，一直得到西安电子科技大学出版社的支持和鼓励，我的学生王莹、张芳同学多次协助校对清样，为此付出了辛勤的劳动；另外，本书在编写过程中，广泛地参阅了有关论著，限于篇幅不一加以注明，作者在此一并致谢。

司存瑞

2001 年 10 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言的特点	1
1.1.1 C 语言是一种不完全的结构化程序设计语言	1
1.1.2 C 语言是一种弱类型的语言	2
1.1.3 C 语言是一种中级语言	3
1.1.4 C 语言的语句简练	3
1.1.5 C 语言的移植性强	4
1.1.6 C 语言自身的不足之处	5
1.2 C 语言程序的格式	6
1.2.1 C 语言中程序、文件、函数三者之间的关系	6
1.2.2 C 语言程序在书写格式上的特点	6
1.2.3 组成 C 语言程序的最小可执行单元和最小模块单元	7
1.2.4 C 语言中多个文件程序中的主函数	8
1.2.5 C 语言程序书写格式与可读性	8
1.3 C 语言的实现	9
1.3.1 如何实现一个 C 语言的程序	9
1.3.2 C 语言程序在编译中的错误类型	10
1.3.3 C 语言中多文件程序如何实现	10
1.3.4 主函数带参数的程序如何实现	11
1.3.5 如何调试一个 C 语言的程序	12
1.4 C 语言词法及其规则	13
1.4.1 C 语言的单词	13
1.4.2 C 语言中的标识符、关键字以及它们的区别	13
1.4.3 C 语言中常用的分隔符及其作用	14
1.4.4 C 语言中空白符的含义及其作用	14
1.5 C 语言的应用	14
1.5.1 C 语言的主要应用	14
1.5.2 C 语言与 C++ 语言的关系	15
第 2 章 Turbo C 2.0 程序设计环境与程序调试技术	16
2.1 Turbo C 的启动与退出	16
2.1.1 Turbo C 的启动	16
2.1.2 退出 Turbo C	17
2.1.3 联机帮助	17
2.2 窗口	18
2.2.1 编辑窗口(Edit window)	18
2.2.2 信息窗口(Message window)	18
2.2.3 监视窗口(Watch window)	19

2.3	菜单与热键	19
2.3.1	打开菜单	19
2.3.2	热键	21
2.4	Turbo C 菜单命令	22
2.4.1	File 菜单	22
2.4.2	Edit 菜单	23
2.4.3	Run 菜单	23
2.4.4	Compile 菜单	24
2.4.5	Project 菜单	25
2.4.6	Options 菜单	26
2.5	文本编辑	31
2.5.1	简单编辑操作	31
2.5.2	块操作	33
2.5.3	查找与替换	34
2.5.4	其它编辑操作	35
2.6	程序调试	37
2.6.1	Turbo C 调试器	37
2.6.2	用于调试程序的菜单命令	38
2.6.3	程序调试技术	42
2.6.4	程序调试实例	47
第3章	C 语言程序设计上机考试应试指南	58
3.1	上机考试系统使用说明	58
3.1.1	上机考试环境	58
3.1.2	上机考试时间	59
3.1.3	上机考试题型及分值	59
3.1.4	上机考试登录	59
3.1.5	试题内容查阅工具的使用	61
3.1.6	考生目录和文件的恢复	64
3.1.7	文件名的说明	64
3.2	上机考试内容	65
3.2.1	DOS 命令操作题	65
3.2.2	程序修改题	72
3.2.3	程序设计题	76
3.3	如何准备计算机等级上机考试	79
3.3.1	了解试卷, 胸中有数	79
3.3.2	勤于实践, 注意积累	80
3.3.3	整理错误, 注意防范	89
3.3.4	查漏补缺, 全面掌握	101
第4章	历年全国计算机等级考试上机考试试题分析	102
4.1	第一卷试题分析	102
4.2	第二卷试题分析	110
4.3	第三卷试题分析	114
4.4	第四卷试题分析	118

4.5	第五卷试题分析	124
4.6	第六卷试题分析	128
4.7	第七卷试题分析	131
4.8	第八卷试题分析	135
4.9	第九卷试题分析	140
4.10	第十卷试题分析	144
4.11	第十一卷试题分析	148
4.12	第十二卷试题分析	153
4.13	第十三卷试题分析	159
4.14	第十四卷试题分析	164
4.15	第十五卷试题分析	168
4.16	第十六卷试题分析	173
4.17	第十七卷试题分析	177
4.18	第十八卷试题分析	183
4.19	第十九卷试题分析	188
4.20	第二十卷试题分析	192
第5章	全国计算机等级考试上机考试模拟试卷	198
5.1	模拟试卷一	198
5.2	模拟试卷二	200
5.3	模拟试卷三	203
5.4	模拟试卷四	205
5.5	模拟试卷五	206
5.6	模拟试卷六	210
5.7	模拟试卷七	213
5.8	模拟试卷八	216
5.9	模拟试卷九	219
5.10	模拟试卷十	222
5.11	模拟试卷十一	225
5.12	模拟试卷十二	227
	全国计算机等级考试上机考试模拟试卷参考答案	231
附录A	全国计算机等级考试说明与大纲	241
附录B	C语言关键字	247
附录C	运算符的优先级与结合性	248
附录D	C库函数	250

第 1 章

C语言概述

C语言是一种具有鲜明特点的结构化程序设计语言。由于这种语言具有许多优点，为编程提供了诸多方便，因此人们比较喜欢使用这种语言。那么，C语言到底有些什么特点？C程序的格式怎样？C程序的词法、规则如何？这些问题，尽管读者在学习C语言程序设计教程时已有一些了解，但在设计程序的过程中，对下面的内容应当十分清楚。

1.1 C语言的特点

1.1.1 C语言是一种不完全的结构化程序设计语言

首先，我们要看到C语言是一种结构化程序设计语言，它具有结构化程序设计语言所必须具有的三种基本结构模式：顺序结构、分支结构和循环结构。因此，可以使用C语言编写出结构化的程序。结构化程序设计方法具有易理解、易调试、易修改等优点，所以程序员都喜欢使用结构化程序设计方法进行程序设计。

C语言又是一种不完全的结构化程序设计语言，这是指：我们首先肯定它是一种结构化程序设计语言，同时又指出它与较严格、较规范的结构化程序设计语言有不同之处。例如，它与PASCAL语言相比较，二者都是结构化程序设计语言，而PASCAL语言要比C语言规范得多，严格得多。正是由于C语言有些不够严格、不够规范的地方，因此，才说它是一种不完全的结构化程序设计语言。

C语言结构化的不完全性主要表现在如下两个方面：一是严格的结构化程序设计语言中是不允许使用goto无条件转向语句的，因为goto语句会破坏结构化。例如，PASCAL语句中就禁止使用goto语句。C语言中允许使用goto语句，但是转向范围受到限制，即在C语言中规定限制使用goto语句。在使用goto语句时，转向范围只能在该函数体内。这就是说，不允许goto语句从一个函数体内转向到另一个函数体，只允许在一个函数体内转向。在实际编程中，应该尽量少用goto语句，最好不使用它。因为使用goto语句即使不会影响结构化，也会降低可读性。一般来说，goto语句仅用在与if语句形成循环语句和在多重循环中从内重循环一次退出到最外重循环。二是严格的结构化程序设计语言中的结构化模块只允许有一个入口和一个出口。而C语言中允许一个函数(函数是C语言中最基本的结构化模块)有多个出口，即允许一个函数中可以使用多个return返回语句，因此会出现多个出口。

上述两点是C语言的结构化不完全的表现，不难看出，这也正是C语言灵活性的表现。为了使用起来灵活方便，C语言对结构化中的某些规定有所“突破”，这也正是C语言

的一个特点。了解了这一点,有利于使用 C 语言灵活地编写程序。

1.1.2 C 语言是一种弱类型的语言

我们说 C 语言是一种弱类型的语言,是指它与 PASCAL 语言相比,在数据类型方面不如 PASCAL 语言那样严格,有些类型可以自动转换,特别是在有些情况下,类型不一致也不报错,因此我们说 C 语言中类型比较弱,故称为弱类型语言。

C 语言中数据类型较弱表现在有些类型可以自动转换。例如, int 型和 char 型之间可以自动转换。这就是说,在一定条件下, int 型可以转换成 char 型, char 型可以转换成 int 型。这就为在 C 语言中实现一个字符常量加上或减去一个整数,或者两个字符常量可以相减等运算提供了方便。下列表达式在 C 语言中是合法的:

```
'a'+2;  
'x'-5;  
'a'-'A';
```

这样一些规定为某些操作带来了方便。例如,设 c 是一个字符变量,将它转换为大写字母(将小写字母转换为大写字母),可用如下表达式

```
c='c'-32;
```

又例如, C 语言中, float 型在机器中一律转换成 double 型,这无疑将提高精确度。还有 short int 型自动转换成 int 型; unsigned char 型和 unsigned short 型自动转换成 unsigned int 型,等等。

这些自动转换一方面给 C 语言编程带来方便,可以实现某些在其它语言中难以实现的操作,另一方面也将带来一些麻烦。由于 C 语言允许某些类型之间自动转换,因此,在许多情况下, C 语言编译系统将不作类型检查,一旦出现了类型错也不报错,这必将给编程带来一定的麻烦。从下面的例子,你将会看到不进行严格的类型检查所带来的麻烦。

```
main()  
{ int a=5;  
  float b=5.5;  
  printf("%d, %.2f\n",b,a);  
}
```

执行该程序输出结果为: 0,0.00

造成莫名其妙结果的原因在于类型的不一致性。因为 printf() 函数的控制串中两个格式符分别是 %d 和 %.2f, 它要求对应的表达式的类型是 int 型和 float 型,而实际上程序中给出的却是 float 型和 int 型。因此,类型不一致,而编译时又不作类型检查,导致上述错误的发生。

可见 C 语言的弱类型给编程带来了方便性和灵活性,同时也带来了一些麻烦。因此,在用 C 语言编程时,要注意类型的一致性问题。例如,在函数调用时,要求形参和实参的类型一致,一般情况下,出现了不一致,也不报错,只是在结果上出问题。克服 C 语言弱类型带来的麻烦,一是要注意类型的一致性问题;二是要在必要时采用强制类型转换的措施,即使用强制类型运算符;三是选用一些进行类型检查的方法。例如,用原型进行函数说明时,调用函数则对形参和实参进行类型检查;用类型定义 typedef 语句定义的类型,

使用时进行类型检查等。

1.1.3 C 语言是一种中级语言

如何看待“C 语言是一种中级语言”，关键在于搞清楚这里所指的“中级语言”的含义。C 语言是一种高级语言，这一点是毫无疑问的，C 语言又是一种低级语言（这一点下面进行解释）。从这个意义上来讲，有人说 C 语言是中级语言，因为它具有高级语言和低级语言两方面的特点，实际上是说 C 语言比高级语言更高级。

C 语言到底具有哪些低级语言的特点呢？第一，C 语言具有二进制位运算的功能，即可对二进制数进行逻辑位操作（包括按位与、按位或、按位求反、按位异或）和移位操作（包含左移和右移）。位操作功能使得 C 语言扩大了应用范围。第二，C 语言具有寄存器操作功能，通过 C 语言编程可实现用 CPU 通用寄存器存取数据，进而大大地提高执行速度。C 语言提供了一种寄存器型的存储类，定义为寄存器类的变量，运行时它有可能被存放到 CPU 的通用寄存器中，一旦被放到寄存器中，存取的速度会比存放在内存中提高很多。因此，人们在编程中常常将一些使用频度较高的变量定义为寄存器类，以求提高执行速度。第三，C 语言具有地址操作的功能。地址操作是指对内存的地址进行操作。由于 C 语言引入了地址这一概念，指针是一种用来存放某种变量地址值的变量。因此，对指针的运算实际上就是对地址值的运算，但是两者还是有不同之处的。指针的使用给构造数据类型变量的操作带来了很大的方便。上述三点是属于低级语言中的功能，而 C 语言都具备。

由于 C 语言既具有高级语言的特点，又具有部分低级语言的功能，则它不仅可用来编写应用软件，又可用于编写系统软件。例如，大家所熟悉的 UNIX 操作系统的大部分程序是用 C 语言编写的；又例如，当前流行的很多数据库系统的源程序和图形系统的源程序都是用 C 语言编写的。

1.1.4 C 语言的语句简练

语句简练是 C 语言的一大特点，主要表现在如下几个方面：

(1) C 语言中类型说明符使用简写字符。例如，整型说明用 `int`，字符型说明用 `char` 等，这相对于 PASCAL 语言更简单些。

(2) C 语言中语句简练的重要表现在于它有许多功能较强的运算符，从而使得语句简练。例如，计算前 10 个自然数的和，可用如下程序：

```
main( )
{ int i=1,s=0;
  while(i<=10)
    s+=i++;
  printf("%d\n",s);
}
```

该程序中使用 `s+=i++`；表达式求和值。这里用到 `++` 运算符，该运算符具有改变变量值的功能，使 `i` 每次增 1，这种功能较强的运算符使得 C 语句更为简练。

又例如，已知两个数，求其最大者，可使用下述程序：

```
main( )
{ int a,b,max;
```

```
a=5; b=8;
max=a>b? a: b;
printf("%d\n",max);
}
```

该程序中使用了三目运算符组成的条件表达式 $a > b ? a : b$ 来求两个数中最大者, 该条件表达式相当于一个 if-else 语句:

```
if(a>b)
    max=a;
else
    max=b;
```

可见使用 C 语言所提供的功能较强的运算符会使语句简练, 程序短小。

(3) C 语言还提供了预处理功能。预处理功能主要包含宏定义、文件包含和条件编译。适当地使用预处理命令会使 C 语言的程序语句简练。例如, 下面程序中使用了宏定义使得程序简练:

```
#define PRD(x) printf("%d\n",x)
main( )
{ int a,b,c;
  a=5;
  b=a+3;
  c=a * b;
  PRD(a);
  PRD(b);
  PRD(c);
}
```

该程序中使用了宏定义, 将一个较复杂的语句定义成一个简洁的标识符(宏名), 这样在书写时就变得简单明了。

1.1.5 C 语言的移植性强

移植性强又是 C 语言的一大特点。C 语言的移植性强是指 C 语言的编译系统可装载在各种大型、中型、小型和微型计算机上, 用 C 语言编写的源程序可运行在不同系统下, 有时只需稍加修改。

C 语言系统的移植性强是因为 C 语言的编译系统是用少于 1 万条代码写成的, 便于修改。因此, 它可以比较方便地移植到各类机器上去。统计资料表明, 不同机器上的 C 编译程序 80% 的代码是公共的。

用 C 语言编写的源程序可在不同系统下运行, 一般只需稍加修改就可以了, 其原因有以下几点:

(1) C 语言提供预处理功能可将与机器相关的因素与主代码分开, 在用于不同机器时只修改与机器相关部分的代码即可, 因此为移植提供了方便。

(2) C 语言编译系统本身不提供输入输出功能, 而输入输出库函数是由支持它的操作系统提供的, 这也为移植带来方便。

(3) C 语言的语义性不强, 一些解释留给具体实现时去规定。

1.1.6 C语言自身的不足之处

C语言具有功能性强、语句简练、编程方便灵活、移植性好等特点,是人们喜欢使用的一种结构化程序设计语言。但在使用C语言的过程中,人们也会发现该语言还存在一些不足,不认识这些不足往往在编程中会出现一些莫名其妙的问题。C语言本身存在的不足有如下几点:

(1) C语言运算符多,使用灵活,存在有难记、难用的不足。标准C语言具有40多种运算符,分为15个优先级,两类结合性。同一种运算符在不同的条件下具有完全不同的功能。例如,星号(*),它作为单目运算符,表示取内容,即取其后面跟的地址值所指向的变量的值;它作为双目运算符,表示乘法操作,即求两个操作数之积。另外,星号与斜线符还可组成注释符(/*, */)等。有的符号既可作为运算符又可作其它符号用。例如,星号(*),前面讲过,它可作运算符使用,又可作说明符使用,在说明语句中,出现在变量名前面的星号表示该变量是指针。又例如,逗号(,)可以是运算符,用它将若干个表达式连成一个逗号表达式,还可以是分隔符,用来作多个变量或多个参数的分隔符。因此,初学C语言时首先要在运算符上下功夫,既要记住它们各自的功能,又要分清它们的区别。

(2) C语言是一种弱类型语言,这为C语言编程提供了方便性和灵活性。但是,C语言对类型检查不够严格,在一些要求类型一致的情况下,出现类型不一致时不予报错,而是进行“自动”转换,造成数据的混乱。因此,在编程中特别要注意使数据类型保持一致。另外,尽量采用检查类型的措施,保证数据的安全性。有关细节在1.1.2节中已有说明。

(3) C语言与其它高级语言一样都具有数组这一数据类型。C语言在定义数组和使用数组时都很方便。但在给数组赋值时要注意C语言不作越界检查,即使超出了数组的上界,仍然可以赋值,这样便会造成数据的混乱,有些有用的数据可能被冲掉。这一点在编程时应该注意,在定义数组时要开辟足够大的空间以防越界。另外,提醒注意,在给数组赋初值时不会出现越界现象,因此,建议对数组多用赋初值的方法。

(4) C语言中引入一些具有副作用的运算符,其目的是为了增强运算符的功能,使得语句简练,但是这种有副作用的运算符在有些情况下会产生二义性。

C语言编译系统允许改变表达式中各个操作数的计算顺序和参数表中各个参数的计算顺序(这样规定有利于在编译中实现优化)。因而,对于没有副作用的运算符来讲,改变计算顺序将不会影响表达式的值。例如,在表达式 $a+b$ 中,先计算 a ,后计算 b ,与先计算 b ,再计算 a ,其结果是一样的。参数表中更是如此。但是,在具有副作用的运算符组成的表达式中,计算顺序不同将会影响表达式的值。例如:

```
int a=5,b;  
b=a-++a;
```

表达式 $a-++a$ 的值将与计算顺序有关。如果先计算 a ,再计算 $++a$,即从左至右计算其表达式值为 -1 ;如果从右至左计算,其表达式值为 0 (在Turbo C中是从右至左计算)。再举个参数表计算顺序不同而使其结果不同的例子:

```
int a=5,b;  
printf("%d,%d\n",a,a=10);
```

在`printf()`函数的参数表 $a, a=10$ 中,如果按从左至右顺序计算, a 的值为 5 , $a=10$ 的值

为 10, 输出结果应为 5, 10; 如果按从右至左顺序计算, $a = 10$ 的值为 10, a 的值也为 10, 输出结果应为 10, 10(在 Turbo C 中是从右至左计算)。可见, 同样一个参数表, 由于计算顺序不同, 造成的结果也不同。

因此, 应注意在不同的编译系统中, 因计算顺序不同将对带有副作用运算符的表达式值和参数的结果产生二义性。在同一个编译系统下, 计算顺序总是一致的; 在没有带副作用运算符参与的表达式和参数表中不会产生二义性问题。

避免二义性的方法是在表达式或参数表中不要出现因计算顺序不同而产生不同结果的情况, 具体做法是将表达式或参数表中相关的表达式用替换量分开书写。例如, 可将上述例中:

```
b=a-++a;
```

写为

```
b=++a;
```

```
b=a-b;
```

同样, 上述的参数表可写为:

```
b=a=10;
```

```
printf("%d,%d\n",a,b);
```

C 语言的上述四点不足都是学习者在编程中应该注意的问题。

1.2 C 语言程序的格式

1.2.1 C 语言中程序、文件、函数三者之间的关系

C 语言的程序是由一个或多个文件组成的, 每个文件是由一个或多个函数组成的。最简单的程序由一个文件中的一个函数组成。

组成一个程序的多个文件, 它们可以独立地存储、修改、复制、删除甚至编译, 但是必须连接在一起才能运行。单个文件一般是不能运行的。

一个文件至少由一个函数组成, 每个函数可以完成一个功能或操作, 函数之间是通过调用关系连接的。函数的调用可以嵌套。

因此, C 语言的程序实际是一个函数串, 即由若干个函数组成, 这些函数可以被放在一个文件中, 也可以被放在多个文件中。在组成一个程序的多个函数中, 仅有一个是主函数, 其函数名为 `main()`。执行一个程序时, 系统先找 `main()` 函数, 找到后从主函数开始执行。在一个程序中, 如果没有主函数, 该程序无法执行; 如果有多个主函数, 该程序也无法执行。主函数在多个函数的文件中, 可放置在任何位置, 在前、在后、在中间都可以。程序的执行与函数在程序中的位置无关。

C 语言程序、文件、函数之间的关系也可以反过来说, 函数组成了文件, 文件构成了程序。

1.2.2 C 语言程序在书写格式上的特点

总的来说, C 语言程序在书写格式上比较自由、灵活, 不像 PASCAL 语言程序要求得

那么严格。

由于C语言的语句比较简练,有的语句含义比较丰富,约定比较多,因此人们都认为C语言程序的可读性比较差。为了弥补这一不足,要求在书写格式上能够有利于提高可读性。

C语言程序由文件组成,文件由函数组成,函数由若干条语句组成,语句由单词组成。

C语言程序书写格式规定:一行可以写一条语句,也可以写多条语句,C语言中每条语句都必须以分号(;)结束。一条语句又可以书写在若干行上。当一条语句写成几行时,在多数编译系统中,在多数情况下,可以不必用续行符。但在有些情况下,需要用续行符(\),例如,有的编译系统要求在宏定义的宏体中,如用多行表示要用续行符;又例如,在printf()函数中的控制串如需要转行时要求加续行符。可见,不同编译系统,对续行符有不同要求。

一般地讲,对C语言程序书写要求有如下几点:

- (1) 如果一条语句写成多行时,应保持单词的完整性,不要将一个单词分写在两行上。
- (2) 在书写一条语句时,每个单词之间一般用一个空格符分隔开,两条语句之间一般用换行符分隔开,即一条语句占一行,这样阅读时方便。
- (3) C语言中花括号({})使用得比较多,一般情况下,左右花括号各自独占一行,并且上下对齐,这样便于检查花括号的成对性。但是在用花括号作为赋初值的初始值表和作为枚举表时,它不必独占一行。一般地讲,用花括号作为函数体定界符号和作为分程序的定界符号时,它最好独占一行。
- (4) 在C语言程序中运用缩格写法对增强程序的可读性十分重要。使用缩格写法可以清楚地表示出语句的层次性。一般要求,同一层次的语句要左对齐,不同层次的语句一般向右缩两个字符。如果循环体内还有复合语句时,复合语句中各语句也应右缩两个字符,用来分清不同层次。

(5) 注释符可用在程序中的任何地方。习惯使用注释符可增加程序的可读性。

建议初学C语言的读者要学会和掌握C语言的一般书写格式,多看一些书写格式较好的程序,逐步学会书写方法。

1.2.3 组成C语言程序的最小可执行单元和最小模块单元

组成C语言程序的最小可执行单元是语句,它表示实现某种功能的一个动作。一条语句是由若干个单词组成的。一个函数又是由若干条语句组成的。一个函数的功能就是由组成它的若干条语句来实现的。语句的种类很多,每种语句有其不同的功能和格式,掌握好各种语句的用法和书写格式是学好C语言编程的重要基础。在C语言程序中,每个语句都必须以分号(;)结束。

组成C语言程序的最小模块单元是函数。函数是C语言中的一个非常重要的概念,是C语言结构化程序设计的基础。C语言程序就是由若干函数构成的函数串,每个函数作为一个独立的模块实现某种功能,组成程序的若干个模块一起完成该程序的功能。函数是一个独立的程序单元,函数之间可以通过指定的方式进行信息交换,函数间的关系是调用关系,一个函数可以被另一个函数调用,但主函数除外。主函数可以自身调用(即递归调用)。一个大程序可以被划分为若干个大模块,每个大模块又可分成若干个小模块,每个小模块

还可以再分为更小的模块,但是最后不可再分的是函数,这便是结构化程序设计方法的多层次(或称多模块)结构方式。

1.2.4 C 语言中多个文件程序中的主函数

在多个文件的程序中,主函数 `main()` 只能有一个。这就是说,在多个文件组成的程序中,只能在某一个文件中包含有主函数 `main()`,而其它文件中不得再有主函数,通常将含有主函数的文件称为主文件。在多个文件组成的程序中只有一个主文件。

多文件的程序中,主函数可以放在任何一个文件中,但只能放在一个文件中。

主文件中可以含有多个函数,主函数是其中的一个,它可以放在文件首、文件尾或文件的中间某处,与存放位置无关。

根据具体实现的需要,主函数可以有参数,也可以没有参数。

主函数无论放在文件的任何位置,当程序被编译连接后,在运行中总是先去执行主函数然后再由主函数去调用其它函数,函数的调用是允许嵌套的。

1.2.5 C 语言程序书写格式与可读性

C 语言程序书写得好会增强可读性,如果书写得不好,则将使程序很难看懂。下面举个例子,请看下面格式一的程序和格式二的程序哪个可读性强。

格式一:

```
main() {int
a,b; printf
("Please input a and b: "); scanf
("%d%d",
&a&b);
printf("a+b= %d\n",
a+b); }
```

格式二:

```
main()
{ int a,b;
printf("Please input a and b: ");
scanf("%d%d",&a&b);
printf("a+b= %d\n",a+b);
}
```

上述两种书写格式写的完全是一个程序,编译时两种格式都不会报错,运行结果会完全一样。但是,很容易看出格式二比格式一要易读得多,可见书写格式对可读性有很大的影响。

下面可以分析一个程序的输出结果。

```
main()
{ int x,y,z;
x=5;
y=7;
```

```
z=-1;
if(x==y)
    if(x>y) z=1;
else z=0;
printf("%d\n:",z);
}
```

如果你认为该程序的输出结果为 0 就错了。该程序的结果应该为 -1。为什么呢？应该认清 else 是第二个 if 语句的，即是 if(x>y) 的。而书写上将 else 与第一个 if 语句对齐而造成了一种错觉，认为 else 是属于第一个 if 语句的。好的书写应该将 else 与第二个 if 语句对齐，可见书写不当会引起错觉。

总的来说，好的书写格式对分析程序是很有益处的。

1.3 C 语言的实现

1.3.1 如何实现一个 C 语言的程序

实现 C 语言的程序与实现其它高级语言的程序一样，都要经过编辑、编译和连接以及运行三个过程。

编辑是将若干条语句按顺序编写成文件。编辑 C 语言程序文件可以用 C 语言编译系统中已有的编辑器，也可用其它通用的编辑器，例如，WPS 等。在使用通用编辑器时，C 语言源程序的文件名应加扩展名 .C。使用编译系统自带的编辑器可不加扩展名，系统自动加扩展名。在编辑过程中，发现有错误可及时修改，将编辑好的程序按指定名字存入磁盘，这样便生成了磁盘文件。

编译和连接实际上是两个过程，但很多编译器常将这两个过程连在一起。编写好的源程序在运行之前必须经过编译和连接过程。编译前先进行宏替换，编译过程就是将源程序翻译成等价的目标代码的过程。编译过程中具体操作如下：

- (1) 将源程序转换成机内码表示的程序，该程序不是机器语言。
- (2) 进行词法分析，辨认出各种单词。
- (3) 根据语法规则进行语法分析，对各种语法成分进行必要的加工处理，生成各种中间语言，最后生成对应的机器代码，即机器语言程序，在这个过程中同时进行代码优化。

连接过程是在生成机器代码以后。在运行之前，将目标程序的各个部分进行连接装配，生成一个可供运行的可执行程序。在这个过程中，考虑地址代换问题，即将逻辑地址（相对地址）转换成对应的物理地址（实际地址），同时还要考虑尽量紧缩所占的内存空间。

在编译连接过程中，发现程序中的错误及时报告，在不断修改错误中完成编译连接过程，最后生成可执行文件（扩展名为 .EXE 文件）。

不同的 C 语言编译系统编译连接的方法是不尽相同的。有的编译系统提供了命令行方式和集成菜单方式两种编译连接方法，例如，Turbo C 编译系统。有的编译系统只提供其中一种方法。

运行是将编译连接好的可执行文件执行的过程，通过运行可执行文件便可获得该程序