

全国计算机等级考试

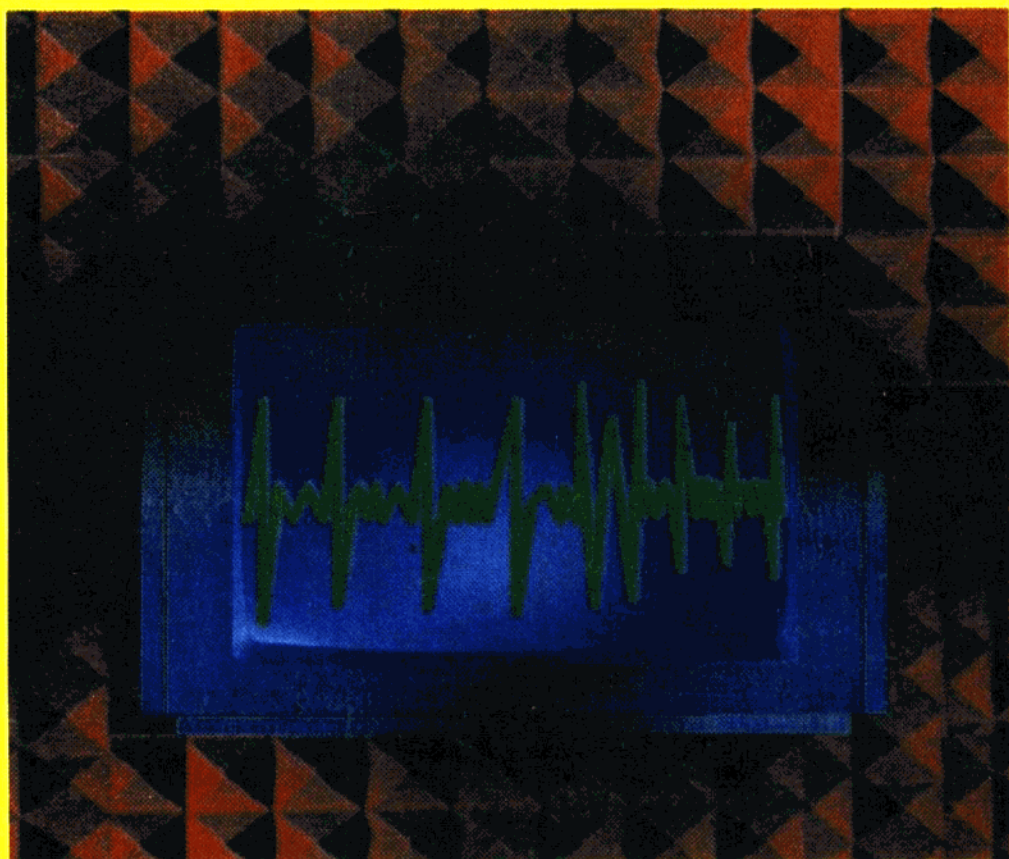
考试指导

(二级 C 程序设计)

国家教委考试中心 组 编

谭 浩 强 主 编

谭 浩 强 编 著



清华大学出版社



第一届全国计算机等级考试 委员会成员名单

主任委员：杨芙清

副主任委员：（以姓氏笔画为序）

朱三元 杨学为 罗晓沛 谭浩强

委员：（以姓氏笔画为序）

王义和	王申康	古天祥	齐治昌	仲革豪	刘淦澄
刘瑞挺	李大友	李克洪	吴文虎	沈钧毅	杨洪
杨明福	林卓然	施伯乐	钟津立	侯炳辉	俞瑞判
张福炎	袁开榜	席先觉	唐兆亮	徐沪生	钱维民
潘桂明	鞠九滨	瞿坦			

秘书长：徐沪生

开展全国计算机等级考试, 为国家经济信息化服务

(代序)

中国科学院院士 北京大学计算机科学系主任 杨芙清
全国计算机等级考试委员会主任委员

当今世界,社会和经济的发展,对信息资源、信息技术和信息产业的依赖程度越来越大,信息化是世界各国发展经济的共同选择。信息产业的发展水平已成为衡量一个国家发展水平和综合国力的重要标志。90年代以来,以计算机、通信、微电子和软件技术为核心的电子信息产业在发展人类的文明,促进国家经济信息化过程中起着非常关键的作用。

今天,一切经济活动都离不开信息,计算机、通信、微电子和软件技术为宏观经济信息的采集、传输、存储、共享、调用、处理、分析和综合等,提供了全新的技术手段。以计算机技术为基础的高新技术的广泛应用,正改变着人们的生产方式、工作方式、生活方式和学习方式,使信息经济财富的增值空间扩大到全球范围,不再受国界的限制。因此,国家经济信息化是世界性的大趋势,中国没有别的选择,只有走国家经济信息化的道路,走与全球信息化融合的道路,与国际接轨,才能在世界经济竞争中有立足之地。

随着计算机的广泛应用,世界各国已采用 EDI(电子数据交换)作为国际经济和贸易往来之主要手段,从根本上改变了国际产业结构和贸易方式。国家经济信息化已成为我国“复关”,并实现我国经济与世界经济接轨的“通行证”。由此可见,计算机在国家经济信息化中的重要性。不会使用计算机甚至就无法进行国际交流。从这种意义上来说,计算机已不是单纯的一门科学技术了,它是跨越国界、进行国际交流、推动全球经济与社会发展的手段,被誉为当今世界的“第二文化”。

进入90年代以来,世界各国竞相发展信息产业,提出一系列推进国家经济信息化进程的高科技发展计划,其中最令人关注是美国的国家信息基础设施(National Information Infrastructure, NII),俗称“信息高速公路”。作为21世纪社会信息化的基础工程,“信息高速公路”将融合现有的计算机联网服务,电视及有线电视的功能,能传递数据、图象、声音、文字等各种信息,其服务范围包括教育、金融、科研、卫生、商业和娱乐等极其广阔的领域,它对全球经济及各国政治和文化都带来重大而深刻的影响。我国也已把加快国家经济信息化提到重要日程。作为国家经济信息化的核心技术,计算机及软件技术将更密切地同人类社会、经济及文化生活联系在一起,不懂得使用计算机,甚至就无法在信息社会中生活、学习和工作。

考虑到中国的实际情况,在实现国家经济信息化的过程中,必须解决全民普及计算机知识及应用技能的问题,必须尽快提高整体计算机应用水平,从而使各行业、各层次的人员,不论年龄,知识背景及专业背景如何,都能掌握和应用计算机,从而解决他们自身专业领域的计算机应用问题,为他们本职工作或专业服务,使之与国家经济信息化的需要相适应。

考察西方各国的情况,这些国家在普及计算机知识,全方位、多层次地培养各行各业计算机应用人员方面,有许多经验值得我们借鉴。其中,很重要的一条就是开展全国范围的定期的计算机各类等级考试。

例如,美国最权威的教育考试中心 ETS(Educational Testing Service)就面向美国社会推出了“计算机文化考试”、“高级就业计算机科学考试”和“专业领域考试”等三类考试。又如美国计算机专业人员认证学会 ICCR(Institute for Certification of Computer Professionals)也实施了有关的认证考试。在英国,由英国计算机学会 BCS(British Computer Society)和 IDPM(Institute of Data Processing Management)分别组织计算机等级考试,并普及到英联邦及其它国家。在日本,自 1969 年开始设立“信息处理技术人员考试”,并已经成为仅次于日本大学全国统一考试的第二大规模全国性考试。

国外的计算机等级考试已有二十多年的历史了,至今方兴未艾。各类考试均有专门的机构长年主持。由于这些考试具有公认的权威性及公正性,因此每次考试参加的人数,多达数万甚至数十万人。不仅应考人员把能否通过这类考试,取得合格证书,作为检验自身计算机技能及择业的重要依据;而且各用人单位也纷纷把是否持有相应的合格证书作为聘用人员的一条重要标准。持有合格证书的人员,当然就在择业、聘用及晋升上具有了有利竞争地位。

令人高兴的是,国家教委考试中心,为了适应我国必须尽快实现国家经济信息化,提高全社会的计算机应用水平,使我国的计算机应用与国际社会接轨的形势的要求,决定自 1994 年起推出全国计算机等级考试。

国家教委考试中心是全国综合性考试管理机构,它承担着高等院校入学统一考试,高等教育自学考试及各种外语考试等多种全国性考试。因此,在全国性考试的组织与管理上,有着丰富的经验,并有遍布全国的考试网点和配套的培训体系。全国计算机等级考试一决定推出,就受到全国各地著名计算机专家和各部门主管领导的热情支持和大力配合;同时得到社会各界的积极反响。一些行业、部门将掌握计算机知识和应用技能列为选拔年轻干部的必备条件之一,因此,全国计算机等级考试为培养年轻干部,提高年轻干部现代化素质提供了良好的机会。此外,全国计算机等级考试既为各行各业用人单位在聘用计算机应用人员方面提供了一个科学而公平的考核标准,又为非计算机专业人员在择业、人才流动、晋升等方面开辟了一条道路。可以想象,随着全国计算机等级考试的推行,不需几年,我国将涌现出大量掌握计算机应用知识和应用技能且精通本行业工作的人才,这必将大大推动我国计算机应用的深入开展,提高全民对国家经济信息化的知识,从而加速我国国家经济信息化的进程,并使我国在经济、贸易、科技、文化等诸方面与国际接轨。

预祝全国计算机等级考试首开成功。

前 言

全国计算机等级考试第二级为程序设计。要求考生掌握计算机的基础知识,并且具有用一种高级语言编写程序的能力。规定考试的语种为 BASIC, FORTRAN, PASCAL, C 四种高级语言和一种微机数据库语言(dBASE III/FoxBASE/FoxPro),考生可从中任选一种。根据目前情况,暂定 BASIC 语言的版本为 MS BASIC(即 IBM PC 用的 BASICA 或 GWBASIC),数据库语言为 FoxBASE。

根据广大考生的要求,全国计算机等级考试委员会决定编写“考试指导”。第二级的考试指导包括 6 本,即“计算机基础知识”、“BASIC 程序设计”、“FORTRAN 程序设计”、“PASCAL 程序设计”、“C 程序设计”、“FOXBASE 数据库系统”。考生可以根据所选择考试的语种选用以上书籍。

应当说明,考虑到多数参加第二级考试的考生已有一定的程序设计知识,因此本书的任务不是帮助读者从头学习以上内容,而是根据大纲的要求系统而简明地介绍应该掌握的内容,由于篇幅关系,也无法容纳过多的程序例题(特别是算法复杂的较长程序)。本书的作用只是帮助考生复习,而不企图代替一本详尽的教材。

建议先系统地、逐章地、仔细地阅读本书,如果对书中介绍的内容都能理解,请不看书本独立地做各章习题以考查自己掌握的程度。如果对某部分的内容比较生疏或感到难以理解,应该重新学习有关教材,可以参阅本书最后列出的参考书籍。

应当说明,本书只包含考试大纲所列出的最基本的内容,而且是提纲挈领式的。有些较深入的叙述或难度较大的问题无法在本书中充分展开,尤其不应理解为:考题全部在本书范围内,或者看完本书就能做出全部考题。考生应能在掌握基本内容的基础上举一反三,能对不同形式、不同难度的考题作出正确的反应。

由于时间仓促,加之水平有限,有不足之处,敬请批评指正。

全国计算机等级考试考试指导(二级)

主编 谭浩强

1995 年 1 月

· V ·

目 录

第1章 C语言的初步知识	1
1.1 C程序的构成	1
1.2 函数的构成	2
1.3 源程序的书写格式和程序风格	3
习题	3
第2章 数据类型及其运算	5
2.1 C的数据类型	5
2.2 常量与变量	5
2.2.1 常量和符号常量	5
2.2.2 变量	5
2.2.3 整型数据	6
2.2.4 实型数据	6
2.2.5 字符型数据	7
2.3 数值型数据间的混合运算	8
2.4 C语言的运算符和表达式	9
2.4.1 C语言运算符	9
2.4.2 C语言表达式的类型	9
2.4.3 表达式的求值规则	10
2.4.4 算术表达式	10
2.4.5 赋值表达式	11
2.4.6 逗号表达式	13
2.4.7 关系表达式	13
2.4.8 逻辑表达式	14
2.4.9 条件表达式	16
习题	16
第3章 基本语句	19
3.1 C语句概述	19
3.2 赋值语句	20
3.3 数据输入与输出	20
3.3.1 字符输入输出函数	20
3.3.2 格式输出函数	21
3.3.3 格式输入函数	23

3.4 程序举例	26
习题	27
第4章 控制语句	29
4.1 if 语句	29
4.1.1 if 语句的三种形式	29
4.1.2 if 语句的嵌套	31
4.1.3 用 if 语句实现选择结构	31
4.2 switch 语句	33
4.3 goto 语句	34
4.4 while 和 do while 语句	35
4.4.1 while 语句	35
4.4.2 do-while 语句	35
4.4.3 while 和 do while 循环的比较	35
4.5 for 语句	36
4.6 break 语句和 continue 语句	39
4.6.1 break 语句	39
4.6.2 continue 语句	39
4.7 循环的嵌套	40
4.8 程序举例	40
习题	42
第5章 数组	47
5.1 数组的定义和引用	47
5.1.1 数组的定义	47
5.1.2 数组的初始化	48
5.1.3 数组元素的引用	50
5.2 数组应用举例	50
5.3 字符数组	53
5.3.1 字符数组的定义	53
5.3.2 字符数组的初始化	54
5.3.3 字符串结束标志	54
5.3.4 字符数组的输入输出	55
5.3.5 字符串处理函数	56
5.3.6 程序举例	58
习题	59
第6章 函数	62
6.1 概述	62

6.2	函数定义的一般形式	63
6.3	函数参数和函数的值	64
6.3.1	形式参数和实际参数	64
6.3.2	函数的返回值	64
6.4	函数的调用	65
6.4.1	函数调用的一般形式	65
6.4.2	函数调用的方式	66
6.4.3	对被调用函数的说明	66
6.5	函数的嵌套调用	67
6.6	函数的递归调用	68
6.7	数组作为函数参数	70
6.8	局部变量和全局变量	73
6.8.1	局部变量	73
6.8.2	全局变量	74
6.9	动态存储变量与静态存储变量	75
6.9.1	变量的存储类别	75
6.9.2	局部变量的存储方式	76
6.9.3	全局变量的存储方式	77
6.10	内部函数和外部函数	78
6.10.1	内部函数	78
6.10.2	外部函数	79
	习题	79
第7章	编译预处理	81
7.1	宏定义	81
7.1.1	不带参数的宏定义	81
7.1.2	带参数的宏定义	82
7.2	“文件包含”处理	84
7.3	条件编译	86
	习题	87
第8章	指针	90
8.1	指针的概念	90
8.2	变量的指针和指向变量的指针变量	90
8.2.1	指针变量的定义	91
8.2.2	指针变量的引用	91
8.2.3	指针变量作为函数参数	93
8.3	数组的指针和指向数组的指针变量	94
8.3.1	指向数组元素的指针变量的定义与赋值	95

8.3.2	通过指针引用数组元素	95
8.3.3	数组名作函数参数	98
8.3.4	指向多维数组的指针和指针变量	100
8.4	字符串的指针和指向字符串的指针变量	101
8.4.1	字符串的表示形式	101
8.4.2	字符串指针作函数参数	102
8.4.3	字符指针变量与字符数组	103
8.5	函数的指针和指向函数的指针变量	105
8.5.1	用函数指针变量调用函数	105
8.5.2	把指向函数的指针变量作函数参数	106
8.6	返回指针值的函数	106
8.7	指针数组和指向指针的指针	107
8.7.1	指针数组的概念	107
8.7.2	指向指针的指针	108
8.7.3	指针数组作 main 函数的形参	109
习题		110
第 9 章	结构体与共用体	113
9.1	概述	113
9.2	定义结构体类型变量的方法	113
9.3	结构体类型变量的引用	115
9.4	结构体数组	115
9.5	指向结构体类型数据的指针	117
9.5.1	指向结构体变量的指针	117
9.5.2	指向结构体数组的指针	118
9.5.3	用指向结构体的指针作函数参数	118
9.6	用指针处理链表	119
9.6.1	链表概述	119
9.6.2	建立链表	120
9.7	共用体	124
9.7.1	共用体的概念	124
9.7.2	共用体变量的引用方式	125
9.7.3	共用体类型数据的特点	125
9.8	枚举类型	126
9.9	用 typedef 定义类型	127
习题		128
第 10 章	位运算	132
10.1	位运算符	132

10.2 位段.....	134
习题.....	136
第 11 章 文件	138
11.1 C 文件概述.....	138
11.2 文件类型指针.....	138
11.3 文件的打开与关闭.....	139
11.3.1 文件的打开(fopen 函数)	139
11.3.2 文件的关闭(fclose 函数)	140
11.4 文件的读写.....	140
11.4.1 fputc 函数和 fgetc 函数(putc 函数和 getc 函数).....	140
11.4.2 fread 函数和 fwrite 函数	141
11.4.3 fprint 函数和 fscanf 函数	142
11.4.4 其它读写函数.....	142
11.5 文件的定位.....	143
11.5.1 rewind 函数	143
11.5.2 fseek 函数和随机读写	143
11.5.3 ftell 函数	143
11.6 出错的检测.....	144
11.6.1 ferror 函数	144
11.6.2 clearerr 函数	144
习题.....	144
附录 I 常用字符与 ASCII 代码对照表	146
附录 II C 语言中的关键字	147
附录 III 运算符和结合性	147
附录 IV C 语言常用语法提要	148
参考文献	152

第 1 章 C 语言的初步知识

1.1 C 程序的构成

下面是一个简单的 C 程序,它包含两个函数。

[例 1.1]

```
main ()                                /* 主函数 */
{int a,b,c;                            /* 定义变量 */
scanf ("%d,%d",&a,&b);                /* 输入变量 a 和 b 的值 */
c=max(a,b);                            /* 调用 max 函数,将得到的值赋给 c */
printf ("max = %d",c);                 /* 输出 c 的值 */
}

int max(x,y)                           /* 定义 max 函数,函数值为整型,x,y 为形式参数 */
int x,y;                               /* 对形参 x,y 作类型定义 */
{int z;                                /* max 函数中用到的变量 z,也要加以定义 */
if (x>y) z=x;
else z=y;
return(z);                             /* 将 z 的值返回,通过 max 带回调用处 */
}
```

应当知道:

1. 一个 C 程序是由一个或若干个函数构成的。程序中至少应包含一个 main 函数。函数是 C 程序的基本单位。C 程序中的函数相当于其它语言中的子程序。用函数来实现特定的功能。因此,可以说 C 语言是函数式的语言,C 语言的这种特点容易实现程序的模块化。

2. 一个 C 程序总是从 main 函数开始执行的,而不论 main 函数在整个程序中的位置如何(main 函数可以放在程序最前头,也可以放在程序最后,或在一些函数之前,在另一些函数之后)。

3. 被 main 函数调用的函数可以是系统提供的库函数(例如 printf 和 scanf 函数),也可以是用户根据需要自己编制设计的函数(例如例 1.1 中的 max 函数)。

4. 一个 C 程序可以由一个文件组成,也可以由若干个文件组成。例 1.1

的程序是由一个文件组成的,在一个文件中包含两个函数。也可以将两个函数分别放在两

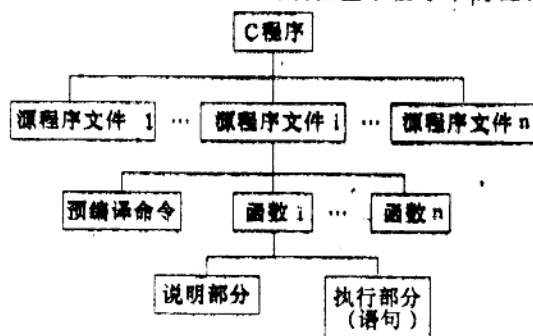


图 1.1 C 程序的结构

个文件中(每一个文件有一个文件名),分别进行编译,然后通过联接(Link)把它们合并成一个可供执行的二进制目标文件,以供运行,图 1.1 表示 C 程序的结构。

1.2 函数的构成

1. 一个函数由两部分组成:

(1) 函数的说明部分,包括函数名、函数类型、函数属性、函数参数(形参)名、形式参数类型。

例 1.1 中的 max 函数的说明部分为:

```
int      max      (x,y)
|         |         |
(函数类型)(函数名)(函数参数)

int      x,y
|         |
(形参类型)(形参)
```

一个函数名后面必须跟一对圆括弧,函数参数可以没有,如 main()。

(2) 函数体,即函数说明部分下面的大括弧|……|内的部分。如果一个函数内有多个大括弧,则最外层的一对|}为函数体的范围。

函数体一般包括:

① 数据说明,对本函数中用到的变量或数组进行定义。如例 1.1 中 main 函数中的“int a, b, c”和 max 函数中的“int z”。

② 执行部分,由若干语句组成。

2. 每个语句和数据定义的最后必须有一个分号,分号是 C 语句的必要组成部分。例如:

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

分号不可少,即使是程序中最后一个语句也应包含分号。

3. C 的 ANSI 新标准对函数的说明部分提出一种新的方式,把函数的形式参数的说明放在函数名后面的“形参表”中,例 1.1 程序可改写为例 1.2 形式。

[例 1.2]

```
int main(void)
{ int max(int, int);
  int a, b, c;
  scanf("%d %d", &a, &b);
  c = max(a, b);
  printf("max = %d", c);
}
```

```
int max(int x, int y)
{ int z;
```

```

if (x>y) z = x;
else z = y;
return (z);
}

```

本程序与例 1.1 相比,有三处区别:

(1) main 函数开头用“int”指定其返回值类型。此“int”也可以不写,不写时隐含表示返回值为整型。关于 main 的返回值,暂可不必深究。

main 后面的括弧中有“void”,表示“形参表是空的”,作用与例 1.1 中 main 后面的括弧是空的一样。

(2) max 函数后面括号中不仅包含形式参数名 x, y, 而且还同时指定其类型。也就是将例 1.1 中 max 函数的说明部分(由二行组成)合并为一行完成。这种方式与 PASCAL 语言类似。

(3) 在 main 函数中增加一行对 max 函数的说明:“int max (int, int);”(也可写成“int max (int x, int y);”)。把它与 max 函数第一行相比较,可以发现二者基本上是一致的,这称为 max 函数的“原型”(function prototype)。它把 max 函数的有关信息(函数名、函数返回值类型、函数参数个数和类型)通知编译系统,以便编译系统检查函数调用是否与“原型”一致,以保证函数的正确调用。

考虑到目前仍广泛使用传统的说明方式(如例 1.1 那样),所以本书中的程序仍以传统方式给出,但读者应了解这种新的说明方式,知道二者间的关系。目前多数 C 系统都允许使用以上两种说明方式。上机都能通过。

1.3 源程序的书写格式和程序风格

1. 定义函数名的部分,后面不要加分号(见例 1.1 main 函数和 max 函数的第一行最后都没有分号)。但对形参的类型说明的最后有分号(见 max 函数第二行“int x, y;”)。

2. C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。为清晰起见,建议一行写一个语句。

3. 为了清晰地表现出程序的结构,建议采用锯齿形程序格式,即将 if 语句的内嵌语句和循环结构中的循环体在书写时向右缩进几列。如多层嵌套则多层缩进。

4. 可以用 /* */ 对 C 程序中的任何部分作注释。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。

习 题

- 1.1 一个 C 程序是由若干个函数构成的,其中必须有一个 main 函数。
- 1.2 一个函数一般由两部分组成:(1) 函数首部, (2) 函数体。
- 1.3 一个函数体的范围是以 { 开始,以 } 结束的。
- 1.4 一个 C 语句最少应包含 分号。
- 1.5 注释部分以 /* 开始,以 */ 结束。

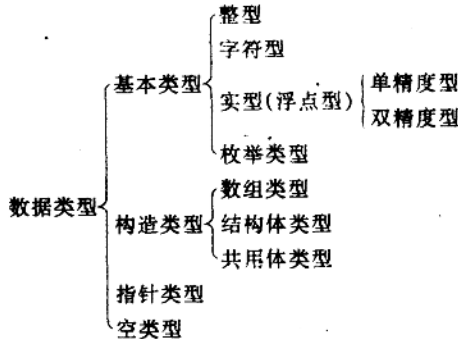
- 1.6 请根据自己的认识, 写出 C 语言的主要特点。
- 1.7 C 语言的主要用途是什么? 它和其它高级语言有什么异同?
- 1.8 请参照本章例题, 编写一个 C 程序, 输出以下信息:

```
*****  
Very good!  
*****
```

- 1.9 编写一个 C 程序, 输入 a、b、c 三个值, 输出其中最大者。
- 1.10 如何上机运行一个 C 程序(以在微机上运行的 Turbo C 为例)。

第2章 数据类型及其运算

2.1 C的数据类型



2.2 常量与变量

2.2.1 常量和符号常量

在程序运行过程中其值不能被改变的量称为常量,常量区分为不同的类型,如 12, 0, -3 为整型常量, 4.6, -1.23 为实型常量, 'a', 'd' 为字符常量。常量一般从其字面形式即可判别。也可以用一个标识符代表一个常量,如可以在一个文件的开头写以下一个命令行:

```
#define PRICE 30
```

用 #define 命令行定义 PRICE 代表常量 30, 此后凡在该文件中出现的 PRICE 都代表 30, 可以和常量一样进行运算。

这种用一个标识符代表一个常量的称为符号常量,即标识符形式的常量,注意,符号常量不同于变量,它的值在其作用域(在本例中为主函数)内不能改变,也不能再被赋值。

习惯上,符号常量名用大写,变量名用小写,以示区别。

2.2.2 变量

其值可以改变的量称为变量。一个变量应该有一个名字,在内存中占据一定的存储单元,在该存储单元中存放变量的值。请注意区分变量名和变量值这两个不同的概念。

和其它高级语言一样,用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符(identifier)。简单地说,标识符就是一个名字。

C 语言规定标识符只能由字母、数字和下划线三种字符组成,且第一个字符必须为字母或下划线。下面是合法的标识符,也是合法的变量名:

```
sum, average, class, day, month, student_name, _above, lotus-1-2-3, basic
```

下面是不合法的标识符和变量名:

M.D.John, \$ 123, #33, 3D64, a>b

注意,大写字母和小写字母被认为是两个不同的字符,因此 sum 和 SUM 是两个不同的变量名。习惯上,变量名用小写字母表示,以增加可读性。

C 语言中标识符的长度(字符个数)无统一规定,随系统而不同。许多系统(如 IBM PC 的 MS C)取 8 个字符,假如程序中出现的变量名长度大于 8 个字符,则只有前面 8 个字符有效,后面的不被识别。例如,有两个变量: student - name 和 student - number, 由于二者的前 8 个字符相同,系统认为这两个变量是一回事而不加区别。

2.2.3 整型数据

1. 整型常量

有以下三种表示形式:

- (1) 十进制形式,如 112, 0, -38。
- (2) 八进制形式,以数字 0 开头,如 0123 表示八进制整数 123,即十进制整数 83。
- (3) 十六进制形式,以 0x 开头。如 0x123 代表十六进制数 123,即十进制整数 291。

2. 整型变量

分为基本型、短整型、长整型、无符号型。

C 标准没有具体规定以上各类数据所占内存字节数,各种机器处理上有所不同,一般以一个机器字(word)存放一个 int 型数据,而 long 型数据的字节数应不小于 int 型,short 型不长于 int 型。

表 2.1 表示的是 PC 机使用的 Turbo C 情况

类 型	类型标识符	在内存中所占位数	数值范围	
基本整型	int	16	-32768—32767	即 $-2^{15}-(2^{15}-1)$
短整型	short[int]	16	-32768—32767	即 $-2^{15}-(2^{15}-1)$
长整型	long[int]	32	-2147483648—2147483647	即 $-2^{31}-(2^{31}-1)$
无符号整型	unsigned[int]	16	0—65535	即 $0-2^{16-1}$
无符号短整型	unsigned short	16	0—65535	即 $0-(2^{16}-1)$
无符号长整型	unsigned long	32	0—4294967295	即 $0-(2^{32}-1)$

2.2.4 实型数据

1. 实型常量

实型又称浮点型,实数有两种表示形式:

- (1) 十进制数形式,它由数字和小数点组成(注意必须有小数点)。0.123, .123, 123.0, 123., 0.0 都是十进制数形式。
- (2) 指数形式,如 123e3 或 123E3 都代表 123×10^3 。但注意字母 e(或 E)之前必须有数字,且 e 后面指数必须为整数,如 e3, 2.1e3.5, .e3, e 等都不是合法的指数形式。

2. 实型变量

分为单精度和双精度两类,在 Turbo C 中,如表 2.2 示。

表 2.2

类 型	类型标识符	在内存中所占位数	能表示数值的有效数字	数值范围
单精度实型	float	32	7	10^{-38} — 10^{38}
双精度实型	double	64	15—16	10^{-308} — 10^{308}

2.2.5 字符型数据

1. 字符常量

有两种形式：一是用单引号括起来的一个单字符，如'a'，'?'，'='等；二是“转义字符”，即以“\”开头的字符序列。常用的转义字符见表 2.3。

表 2.3

字符形式	功 能
\n	换行
\t	横向跳格(即跳到下一个输出区)
\v	竖向跳格
\b	退格
\r	回车
\f	走纸换页
\\	反斜杠字符“\”
\'	单引号(撇号)字符
\"	双引号(")字符
\ddd	1 到 3 位 8 进制数所代表的字符, 如 \123, 表示 8 进制数 123
\xhh	1 到 2 位 16 进制数所代表的字符, 如 \x21, 表示 16 进制数 23

在将转义字符赋给字符变量时, 也需要在其二侧加单引号, 如:

```
char c = '\n';
```

2. 字符串常量

用一对双引号括起来的字符序列, 如"abc\n"。

请区别字符变量和字符串常量。a是字符常量, a是字符串常量。在内存中, 字符常量占一个字节, 而对字符串常量, 则在字符串的后面加一个“字符串结束标志”, 以便系统据此判断字符串是否结束。用“\0”(即 ASCII 代码为 0 的字符, 即“空字符”)作为“字符串结束标志”。例如, "a"在内存中占两个字节, 即:

a	\0
---	----

。

3. 字符变量

用来存放一个字符常量, 字符变量用 char 来定义。如:

```
char a, b;
```

```
a = 'O'; b = 'K';
```