

C 程序设计与错误分析

刘振安 苏仕华 周淞梅 编著

中国科学技术大学出版社

1993·合肥

内 容 简 介

本书以实例为兰线,以通俗的语言和简要的内容阐述了C语言的程序设计方法,同时结合实例,介绍了使用集成环境编辑完整的实用程序(包括头文件的编制、多个C语言文件及工程文件的编制等),查错及调试程序的方法,并对C语言编程错误进行较系统的分析、介绍了预防与排除的方法,本书还配有课程设计实例及习题以加深对C语言的理解。

本书取材新颖、内容丰富,阐述系统,不同领域的读者均可有所受益。可作为大专院校的教材,培训班教材,自学教材及工程技术人员的参考书。

(皖)新登字08号

C 程序设计与错误分析

刘振安 苏仕华 周淦梅 编著

*

中国科学技术大学出版社出版

(安徽省合肥市金寨路96号,230026)

合肥丰航彩印厂印刷

安徽省新华书店发行

*

开本: 787×1092/16 印张 16.5 字数 410千

1993年8月第1版 1993年8月第1次印刷

印数 1-8000册

ISBN 7-312-00528-4/TP·69 定价 7.50元

前 言

C语言是一种通用的程序设计语言，它既不是“非常高级”的语言，也不是“低级”语言，而通常被称为“中级”语言，但并不意味着C语言功能差，难以使用。C语言的通用性和无限制性使得它对许多程序设计来说都比想象中的功能的语言更加通俗，更加有效。目前C语言已用于各个方面的程序设计，无论是设计系统软件（操作系统，编译系统等），还是应用软件（图形处理），数据处理（如企业管理）以及数值计算等都可以很方便地使用C语言。

C语言的特点是多方面的。简单说来，它用如下几个特点：

- (1) C语言吸取了汇编语言的精华，使C语言对高级语言来讲是“低级”语言。由于它具有描述准确和目标程序质量高的优点，所以有很强的生命力。
- (2) C语言继承和发扬了高级语言的长处，使C语言相对汇编来讲又是“高级”语言。
- (3) C语言的规模适中，语言简洁，其编译程序简单而紧凑，它在运行时所需要的支持少，占用的存储空间也小。
- (4) C语言的可移植性好，这是指程序从一个环境不加或稍加改动就可搬到另一个完全不同的环境上运行。汇编程序因依赖机器硬件，所以根本不可移植，而一些高级语言，如FORTRAN等编译的程序也是不可移植的。

由于上述几个突出优点而使越来越多的人加入到学习和研究C语言的行列之中。而C语言以它的格式自由，限制较少，语句简洁等特点更使学习者爱不释手。

本书有如下特点：

1. 以实例为主线，以通俗的语言和简要的内容阐述了C语言的程序设计方法。
2. 理论联系实际。书中列举例题简易，针对性强。实例由浅入深，有一定工程背景，能起到学以至用的效果。
3. 取材新颖，内容丰富，阐述系统。还精选了一些综合应用实例以加深对C语言的理解。
4. 本书以Turbo C为集成开发环境实验手段，不仅给学习者提供了极大的方便，还使学习者对编辑、编译、查错及连接运行C程序有了完整的认识，从而编制自己的实用程序。
5. 介绍了C语言头部文件及多个文件的编制和调试技术，使读者能很快掌握C语言的实用技术，这在C语言程序设计的教材中还有一种尝试。
6. 通过作者自己的经验，较系统地介绍了C语言编程错误、错误分析方法及程序测试方法，以使读者尽快入门并掌握编制完整的C语言应用程序，这不仅对初学者大有好处，对具有一定经验的C语言程序设计者也大有益处。
7. 结合实例，介绍了使用集成环境及调试程序的方法。
8. 介绍了C语言与其它语言的接口。
9. 配备一定数量的基础习题和课程设计实例。

本书共分十三章。第一章介绍C语言程序的编辑、编译和运行；第二章介绍基本的

数据类型和表达式；第三章在介绍简单程序设计的基础上，又介绍了 Turbo C 集成开发环境及调试程序的方法；第四章介绍逻辑运算和分支程序；第五章介绍循环程序设计；第六章介绍函数与变量类型；第七章介绍构造类型（数组和指针）；第八章介绍结构类型；第九章介绍文件；第十章介绍 Turbo C 使用，集成环境下的 C 程序基本开发方法，建立和运行含有多个源文件的 C 程序，Turbo C 库函数、配套工具、实用于程序及 C 语言头部文件等等；第十一章介绍课程设计实例；第十二章介绍查错的一般方法；第十三章介绍 C 语言接口。在教学中，可以把第十章和第十二章的部分内容穿插到前面的教学中去，它们一方面是为了实验，另一方面是为了克服编程错误，所以大部分内容要学生自己去体会并在编程时进行实践。第十三章则不一定讲授，因为这一章是为了拓宽应用面。

本书第 1~2、4~6 章由苏仕华执笔；第 3 章由苏仕华、周淞梅执笔；第 8~9 章由苏仕华、刘振安执笔；第十章由周淞梅、刘振安执笔；第 7、11~13 章由刘振安执笔；最后由刘振安修改定稿。

在本书的写作过程中，得到许多领导的大力支持。安徽大学副校长、程慧霞教授，北京工业大学副校长沈兰荪教授，我校计算机系主任陈国良教授审阅了书稿；中央候补委员、科技大学常务校长汤洪高书记也给予大力支持。特此表示感谢并对被引用的资料的作者再次表示感谢。

本书是在谷校长的鼓励下，为普及我国的计算机教育事业而向读者提供的第三部著作。我们曾收到很多读者及同行的来信，给予我们很多支持和鼓励，使我们受益非浅。在此特向诸位表示感谢，欢迎广大读者一如既往地关心我们，提出宝贵的意见。

因我们才疏学浅，错误之处在所难免，敬请读者和同行批评指正。

作者

1993 年 8 月于合肥

目 录

第一章 C 语言概述	1
1.1 C 语言的发展过程	1
1.2 C 语言的特点	1
1.3 简单的 C 程序分析	2
1.4 C 语言程序的编辑、编译和运行	6
习题 1	7
第二章 基本的数据类型和表达式	8
2.1 标识符和变量	8
2.2 基本数据类型	9
2.3 常量	10
2.4 表达式	12
习题 2	13
第三章 简单程序设计	14
3.1 语句综述	14
3.2 赋值语句	16
3.3 数据输出	17
3.4 数据输入	21
3.5 Turbo C 集成开发环境及其使用	24
3.6 集成调试程序	31
3.7 程序举例	33
习题 3	34
第四章 逻辑运算和分支程序	35
4.1 关系运算	35
4.2 逻辑运算	36
4.3 分支程序设计	36
4.4 条件运算符	40
4.5 switch 语句	42
4.6 goto 语句	45
习题 4	46
第五章 循环程序设计	47
5.1 while 语句	47
5.2 do—while 语句	48
5.3 for 语句	48

5.4	break 语句	54
5.5	continue 语句	56
	习题 5	58
第六章	函数与变量类型	59
6.1	函数	59
6.2	变量类型	68
6.3	变量初始化	74
6.4	C 预处理器	75
	习题 6	79
第七章	构造类型——数组和指针	80
7.1	数组	80
7.2	指针	87
7.3	Turbo C 动态分配函数	91
7.4	指针与数组	94
7.5	指针与函数	101
7.6	命令行参数	104
7.7	指针函数	106
7.8	函数指针	107
7.9	指向指针的指针	110
	习题 7	111
第八章	结构类型	112
8.1	结构定义及其变量的初始化	112
8.2	结构数组	117
8.3	结构指针	121
8.4	引用自身的结构	126
8.5	字段结构	130
8.6	联合	131
8.7	枚举	135
	习题 8	137
第九章	文件	137
9.1	文件概述	137
9.2	文件的打开与关闭	138
9.3	文件的读写	141
9.4	文件的定位	151
9.5	出错的检测	153
9.6	文件输入输出小结	154
	习题 9	154
第十章	Turbo C 使用及实例	155
10.1	菜单命令及其使用	155

10.2	集成环境下的 C 程序基本开发方法	168
10.3	建立和运行含有多个源文件的 C 程序	174
10.4	Turbo C 库函数	180
10.5	Turbo C 的配套工具及实用子程序	180
10.6	C 语言头部文件	181
10.7	TC 的命令行开关	185
10.8	TCC 直接编译和连接 Turbo C 程序	186
第十一章	课程设计实例	189
11.1	综合实例	189
11.2	仿真单片微机	199
第十二章	编程中的常见错误及排除方法	201
12.1	常见的错误类型	201
12.2	错误的种类与预防	215
12.3	错误的检出与分离	225
12.4	程序的测试	229
第十三章	C 语言接口	233
13.1	C 语言与 UNIX 系统的接口	233
13.2	C 语言与 FORTRAN 语言的相互调用	239
13.3	C 语言与 FORTRAN 语言数据文件共享	244
13.4	Turbo C 与 DBASE III 接口	245
13.5	C 语言与汇编语言的接口	248
主要参考文献		252

第一章 C 语言概述

1.1 C 语言的发展过程

C 语言是 70 年代初期美国贝尔 (Bell) 实验室 Dennis M.Ritchie 设计的一种程序设计语言。正式发表于 1978 年。

1970 年, Ken Thompson 在早期的编程语言 BCPL 的基础上, 开发了一种新的语言, 取名叫“B”。1971 年开始, Dennis M.Ritchie 在“B”的基础上, 开发了第一个 C 编译程序, 1972 年开始使用之后, C 语言经过多次改进, 主要是在贝尔实验室内部使用, 直到 1975 年用 C 语言写的 UNIX 操作系统第六版公诸于世后, C 语言才举世瞩目。1978 年 Brian Kernighan 和 Dennis M.Ritchie 在 C 程序语言 (The C Programming Language) 一书中对 C 语言作了详尽的描述。随着微型计算机的日益普及, 大量的 C 语言工具得以问世。然而, 没有统一的标准, 这些工具也有不一致的地方, 为了改变这种情况, ANSI 于 1983 年专门成立一个委员会, 为 C 语言制定了 ANSI 标准。TURBO C 不仅提供了一个集成开发环境, 同时按传统方式提供了命令行编译程序版本以满足不同用户的需要。

1.2 C 语言的特点

C 语言是一种通用的程序设计语言, 它既不是“非常高级”的语言, 也不是“低级”语言, 而通常被称为“中级”语言, 但这并不意味着 C 语言功能差, 难以使用。正是 C 语言的通用性和无限制性, 才使得它对许多程序设计来说都比想象中的功能的语言更加通俗, 更加有效。目前 C 语言已用于各个方面的程序设计, 无论是设计系统软件 (操作系统, 编译系统等), 还是应用软件 (图形处理), 数据处理 (如企业管理) 以及数值计算等都可以很方便地使用 C 语言。

C 语言的特点是多方面的。简单说来, 它有如下几个特点:

(1) C 语言吸取了汇编语言的精华, 使 C 语言对高级语言来讲是“低级”语言。汇编语言是一种面向机器的程序设计语言, 尽管它的编程相对高级语言来要麻烦得多, 但由于它具有描述准确和目标程序质量高的优点, 所以汇编语言仍然有很强的生命力。

① C 语言提供了对位、字节以及地址的操作, 使程序可以直接对内存, 指定寄存器

进行操作。

② C 语言吸取了宏汇编技术中的某些灵活的处理方法, 提供宏代换 `#define` 和文件蕴含 `#include` 的预处理命令。

③ C 语言能很方便地与汇编语言连接。C 程序中引用汇编程序与引用 C 语言函数一样。这为某些特殊功能程序的设计提供了方便。

(2) C 语言继承和发扬了高级语言的长处, 使 C 语言相对汇编来讲又是“高级”语言。

① 吸取了 ALGOL 的分程序结构思想, C 程序中, 可用一对花括号 “{}” 把一串语句括起来而成为复合句 (分程序), 在括号内可定义变量。它还继承了 PASCAL 的数据类型, 提供了相当完备的数据结构。

② 吸取了 FORTRAN 语言的模块结构思想, C 程序中, 它的每一个函数都是独立的, 可以单独编译, 对设计一个大的程序来说, 有利于分工编程和调试。

③ C 程序中的任何函数都允许递归, 这样对某些算法实现起来就十分方便, 清晰。

(3) C 语言的规模适中, 语言简洁, 其编译程序简单而紧凑。C 语言在表示上尽可能地简洁 (比如用一对花括号 {} 代替 `Begin__End`, 运算符尽量缩写等), 语言的许多成份都是通过显示函数调用来完成的, 比如 C 语言中没有 I/O 设施, 也没有并行操作等; 另一方面, 它在运行时所需要的支持少, 占用的存储空间也小。

(4) C 语言的可移植性好, 这是指程序从一个环境不加或稍加改动就可搬到另一个完全不同的环境下去运行。汇编程序因依赖机器硬件, 所以根本不可移植, 而一些高级语言, 如 FORTRAN 等编译的程序也是不可移植的。

C 语言的优点很多, 但也有些不足之处。例如运算符优先级太多, 不便记忆, 有些还与常规约定有所不同; 类型检验较弱, 转换比较随便, 所以不太安全。尽管如此, C 语言仍不失为一个实用的通用程序设计语言。由于上述几个突出优点, 使越来越多的人加入到学习和研究 C 语言的行列之中。但也有些人认为 C 语言是专业程序设计人员的语言。其实这种看法是不全面的, 能学好 COBOL, BASIC 的人同样能学习好 C 语言。而 C 语言以它的格式自由, 限制较少, 语句简洁等特点, 使学习者爱不释手。

1.3 简单的 C 程序分析

1.3.1 C 语言程序结构

用 C 语言编写的程序称为 C 语言源程序, 简称 C 程序。C 程序一般是由一个或若干个函数组成, 而这些函数可以保存在一个或几个源程序文件中, 这些文件都以 .C 作为文件扩展名。在组成一个程序的若干函数中必须有一个且只能有一个名为 `main` 的函数称为主函数, 在运行 C 程序时总是从 `main` 函数开始执行。一个函数在其名字之后一定要有一对圆括号, 圆括号中的参数可有可无。我们首先看一个简单的 C 程序。

【例 1.1】 打印字符串

/* 功能: 打印字符串 */

`main()`

```

{
    printf( " Hello! How are you ? " );    /* 打印字符串 */
}

```

这是一个完整的 C 程序，程序中，以 “/ *” 开头到 “* /” 结尾之间的内容是注释，注释帮助阅读、理解及维护程序，在编译时，注释部分被忽略，不产生目标代码。注释在 C 程序中是很有用的，对一个好的程序设计者来说，应该在程序中使用注释来说明整个程序的功能和注意事项及有关算法等。

在程序 1.1 中有一对花括号，类似于 Pascal 语言中的 “Begin__End”，可以看作程序体括号，也可以用来括起任何一组语句，从而构成一个复合句(或叫分程序)。要注意在一个函数中至少有一对花括号。C 程序的一般函数或 “main()” 之后应有一个 “{”，在函数的最后应是一个 “}”，在一个 C 程序或一个函数中，“{” 和 “}” 必须是成对出现。printf(“Hello! How are you?”); 语句是一个函数调用，调用名叫 printf 的库函数，括号内双引号括起来的是所带的参数，即要打印的内容。printf 是标准输出函数，对应于输出设备终端显示器，上述语句表示要在终端输出字符串

Hello! How are you?

1.3.2 C 函数简述

C 函数分为两类。一类是系统本身提供的库函数(标准函数)，在编程时只要在需要的地方写上函数名，带上参数即可调用库函数，一般情况下要在主函数之前加上相应的蕴含函数库名。比如要调用数学库函数，就要在 main() 之前加上 #include <math.h>，C 语言有非常丰富的库函数；另一类叫做自定义函数，程序设计人员可根据编程的需要自行设计一段程序完成一个特定的功能，它相当于 Fortran 中的子例程子程序或函数子程序，等价于 Pascal 中的函数或过程。C 语言函数的使用简单，方便，执行效率高。在 C 语言程序设计中，要形成良好的设计风格，即用多个小函数或多个小程序构成一个大程序，每一个小函数或小程序完成一个独立的功能，单独编译和调试。请看下面的例子：

【例 1.2】 求 $\text{bin}(n, k) = n! / (k! * (n-k)!)$

```

main()
{
    int  n, k;
    n=8;
    k=5;
    printf( "bin(n, k) = %d\n", fac(n) / (fac(k) * fac(n-k)) );
}

fac(m)    /* 求 m 的阶乘，即 m! */
int  m;
{
    int  i, h;
    h=1;
    for (i=1; i<m; ++i)

```

```

    h = h * i;
    return(h);
}

```

以上是个求阶乘的函数 $\text{fac}(m)$ 及调用它的主程序 (主函数)。主函数中要求函数 $\text{bin}(n, k) = n! / (k!(n-k)!)$ 调用了 3 次即 $\text{fac}(n)$, $\text{fac}(k)$, $\text{fac}(n-k)$ 。从上例可以看出, 每一个函数都有基本相同的形式:

函数类型说明 函数名 (参数表, 如有的话)

参数说明, 如有的话

```

{
    变量说明
    语句部分
}

```

函数可按任何顺序出现, 且可出现在一个源程序文件或多个源程序文件中。函数定义中不可缺少部分是:

函数名 ()

```

{
}

```

其它部分根据需要来确定有无。分析函数的定义形式, 可以把定义分为两部分, 即函数说明部分和函数体。

(1) 函数说明部分: 函数类型说明定义函数返回值的数据类型。C 程序所使用的基本数据类型有 `int` (整型)、`char` (字符型)、`float` (浮点型)。C 程序里没有类型说明的函数隐含为整型。在 C 程序里, 子程序和函数是一个意思, 都称作函数。

函数名是识别函数的名字。有效的名字是以英文字母 (`a~z`, `A~Z`) 或下划线 (`_`) 开始的英文字母、数字及下划线组成的字符序列。一般是以前 8 个字符为准。此外, 在 C 程序里, 函数名用大写字母和小写字母被认为是两个不同的函数。

参数表在函数名后的圆括号 () 内。这里的参数是形式参数, 简称形参 (亦称哑元)。当该函数被调用时, 形参将被实在参数 (亦称实元) 所替换, 这种替换常叫做哑实结合。函数也可以没有参数, 但函数名后的圆括号 () 不能省略。

(2) 函数体: 是需要完成功能的处理部分。

函数体从花括号 { 开始, 直到与此对应的花括号 } 为止。

变量说明通常接在 { 后面, 基本类型有 `int`、`float` 和 `char`。接在变量说明后的是语句部分。函数体的最后是 }, 表示该函数到此结束。

1.3.3 基本的输入与输出

输入输出设施不是 C 语言的一部分, 而是以标准函数形式提供。程序在引用库函数的每一个源程序文件的开头处含有一行: `#include <stdio.h>`。scanf、printf 函数除外。但也有的 C 编译要求用 `#include <stdio.h>` 语句。

文件 `stdio.h` 定义了 I/O 库所用的某些宏和变量。使用 `#include` 把 `stdio.h` 文件包含

进来,一起编译。下面简要介绍格式化输入输出函数 scanf()和 printf()。

C 语言的标准库中提供了两个很有用的格式化输入输出标准函数,它类似于 FORTRAN 中的 READ 和 WRITE 语句,为程序员实现各种格式输入输出提供了方便。

格式化输出函数 printf 的形式如下:

```
printf(控制字符串, 参数 1, 参数 2 ...);
```

printf 的功能是按照控制字符串,将参数进行转换,按格式在标准输出设备上输出。在控制字符串中包含两种字符:一种是普通字符,将原样输出;另一种是格式符,它是以“%”开头紧跟格式字符,说明其对应参数的输出格式,先介绍最常用的四种格式符:

d——将参数按十进制形式输出。

c——将参数看作单个字符输出。

s——将参数所指出的字符串输出,字符串以空字符为终止符。

f——将参数按浮点数形式输出。

例如有下面语句

```
x1 = 1234;
```

```
x2 = 2345;
```

```
printf("x1 = %d x2 = %d\n", x1, x2);
```

其中:“x1 = %d x2 = %d\n”是控制字符串, x1、x2 是参数,控制字符串中的 x1 =、x2 = 和 \n(回车换行)都是普通字符,按原样输出,两个 %d 是格式说明符,依次说明 x1, x2 均按十进制(整型)形式输出。上述语句执行后的输出结果为:

```
x1 = 1234      x2 = 2345
```

格式化输入函数 scanf,它提供的格式与 printf 类似,其格式为:

```
scanf(控制字符串, 参数 1, 参数 2, ...);
```

scanf 函数实现从标准输入设备(终端)上按控制字符所规定的格式输入数值或字符,并将输入内容存在参数所指定的单元中。对于参数的书写,程序设计人员要特别注意,在 printf 中的参数是给出要输出值的变量名,而 scanf 中的参数是要给出接收数据的变量地址。例如语句

```
scanf("%d %d", &x, &y);
```

该语句实现从终端输入二个十进制数分别赋给 X 和 Y。这里 &X 和 &Y 表示 X, Y 的地址。在输入二个十进制数之间用空格分隔。

现举一个例子说明之。已知三角形的边长分别为 A、B、C,求三角形的面积。

【例 1.3】 已知三边,求三角形的面积

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

```
#include <math.h>
```

```
main()
```

```
{
```

```
float a, b, c, area();
```

```
printf("请输入三角形的三边边长:\n");
```

```

scanf("%f%f%f\n", &a, &b, &c);
printf(" 三角形面积为: %f\n", area(a, b, c));
}
area(a, b, c)
{
    float  s;
    s=(a+b+c);
    return (sqrt(s * (s-a) * (s-b) * s-c));
}

```

上述程序中出现的 scanf 函数功能是实现从终端输入三个浮点数分别赋给变量 a, b, c; 第一个 printf 显示一个汉字串, 第二个 printf 除了显示汉字串外还输出一个函数 area 的调用值。在程序中用到了汉字, 这可以在汉字操作系统支持下运行。但更方便的是使用我们生产的“星河”汉卡。因为星河汉卡不需要汉字操作系统的支持, 它在西文系统里运行, 所以西文软件可以直接使用汉字。建议在此系统下, 给程序写详细的中文注释。

1.4 C 语言程序的编辑、编译和运行

Turbo C 提供了一个 C 程序的集成开发环境。在此集成开发环境内可以编辑、编译、连接运行 C 程序。为此, 在文件 tc.exe 所在的目录下键入 tc 并按 Enter 键, 就能启动 Turbo C 进入集成环境。

在本书中, 用 Ctrl-Y 表示同时按下 Ctrl 和 Y 键; Ctrl-KY 表示同时按下 Ctrl 和 Y 键, 都放开后再按下 Y 键。有的书把它记做 Ctrl+KY。ALT+X 的含义与此同理。

1.4.1 源程序的输入与编辑

当你编写好 C 程序后, 即可按上述方法进入集成环境, 再按 Alt-e 进入 Edit 全屏幕编辑器工作状态, 这时就可以输入、编辑、修改你的程序; 当你的源程序已存在磁盘上而想编辑修改时, 你可以首先用 Alt-F 选择 File 菜单的 File / Load, 按 Enter 键, 出现 Load File Name 提示框, 键入你要进行编辑、编译连接、运行的源文件名, 再按 Enter 键将磁盘上的源程序文件装入内存, 编辑修改可按 Alt-e 进入 Edit 状态。

1.4.2 源程序的编译、连接和运行

你的源程序编辑修改完成后, 可按功能键 F2 把程序存入磁盘。如程序是装入的, 即存入原来的文件中; 如是新输入的, 你可以打入一个文件名并加上 .C 作为扩展名或不打入文件名而直接按 Enter 键, 这时源程序就被存入一个叫 noname.c 的文件中。按 Alt-R 进入 Run 菜单, 可选取 Run / Run 并按 Enter 键, 便可编译连接运行你的 C 程序。运行完毕立刻返回到主菜单。如若程序有错, 编译程序可指出错在什么地方, 等修改完后再进

入 Run 菜单操作。

1.4.3 查看运行结果

在连接运行时运行结果是昙花一现，看不清楚，若想要观察清楚可按 Alt-F5 键，看完后按任意键返回主菜单或者先按 Alt-f 进入 File 菜单，选择 File / Os shell 并按 Enter 键，临时返回 DOS 系统，这时可看到运行结果，还可以做一些其它工作。键入 Exit 立即返回 Turbo C 集成开发环境。

11.4.4 编辑命令的速成指南

TC 的常用的一些编辑命令有：

在正文中移动光标，使用 ↑、↓、←、→ 和 PgUp / PgDn 等键。

删除光标所在行用 Ctrl-Y；

删除光标所在单词用 Ctrl-T。

删除一个字符可用 Ctrl-G 键（光标处的）或用 Del 键（光标前的）。

定义一个字块用 Ctrl-KB（块头标记）和 Ctrl-KK（块尾标记）。

移动一块到光标所在处用 Ctrl-KV；

拷贝一字块到光标所在处用 Ctrl-KC；

删除一字块用 Ctrl-KY；

取消块标记用 Ctrl-KH。

更详细的操作命令及 TC 集成环境的使用说明等，见第三章的介绍。

习 题 1

- (1) 进入 TRUBO C 集成环境并练习编辑命令。
- (2) 编辑一个输出“25+75=100”的程序。
- (3) 编辑一个对话程序，例如：

李明：HOW ARE YOU, 王莹。

王莹：FING, THANK YOU AND YOU ?

- (4) 编辑一个计算 $y = 2x + 8$ 的程序并运行之。
- (5) 给第 (4) 题的程序加上注释。

第二章 基本的数据类型和表达式

①

2.1 标识符和变量

2.1.1 标识符

所谓标识符，实际上就是一个字符序列。在 C 语言中，标识符用来标记常量、变量、数据类型、函数及程序的名字。在 C 语言中构成标识符必须符合下列语法规则：

- (1) 以字母或下划线“_”符中任一字符打头；
- (2) 在第一个字符之后，可以是任意的字母、下划线或数字组成的字符序列，这个序列可以是空串。

C 语言中的标识符可以分为下述三类。

(1) 关键字

关键字是用来说明 C 语言中的某一固定含义的字。C 语言中一共有如下 30 个关键字：

int, char, float, double, short, long, unsigned, struct, union, auto, extern, register, static, typedef, goto, return, sizeof, break, continue, if, else, do, while, switch, case, default, enum, for, void, entry.

这些关键字为 C 语言专用符，不得赋予其它含义，C 语言中的习惯是用小写字母，所有这些关键字也都是由小写字母构成的。

(2) 特定字

特定字是具有特定含义的标识符，主要有如下 7 个：

define, include, undef, ifdef, ifndef, endif, line.

它们主要用在 C 语言的预处理程序中，这些标识符虽然不是关键字，但由于这些字赋予特定含义，所以人们习惯把它看作是关键字。因此在程序中不能把这些特定字当作一般标识符使用。

(3) 一般标识符

一般标识符通常是用户根据程序中的需要自己按照标识符规则定义的标识符。这些标识符程序员可根据所标记的内容的含义用英文或汉语拼音来表示，或根据习惯符号来表示。下面的字符串为合法的标识符：

dw, name, f_1, rec_2, x, y, z11, area, _age

而下面的字符串都是不合法的标识符：

2a, -xy, name*, no#, a/b

C 语言中大小写字母是有不同的含义的，例如 name 和 NAME 就代表不同的标识

符。C 语言中规定标识符长度不限，但只有前 8 个字符有效，所以在定义时要注意前 8 个字符不能出现重名。对一个可读性好的程序，其中标识符必须选择恰当，取名统一规范化，使读者一目了然。

2.1.2 变量

数据是计算机程序处理的主要对象，在程序中每项数据不是常量就是变量，它们之间的区别仅在于程序执行过程中变量的值可以改变，而常量是不能改变的。变量就是一般的标识符，用来存储各种类型的数据，以及指向存储器内部单元的指针。

所有的变量在使用之前必须说明，所谓说明是指出该变量的数据类型、长度等信息。说明由类型和具有该类型的变量表组成。如：

```
int lower, upper, step;  
char c, name[16];
```

变量可按任何方式分布在若干个说明中，上述说明同样可以写成：

```
int lower;  
int upper;  
int step;  
char c;  
char name[16];
```

后一种形式会使源程序冗长，但便于给每个说明加注释，也便于修改。上面说明的 lower、upper、step 是整数类型；c 是字符类型；name 是字符数组，可存放 16 个字符。

2.2 基本数据类型

数据类型是语言中非常重要的一个概念，它把一个语言所处理的对象按其性质不同分为不同的子集，对不同的类型规定不同的运算。Turbo C 语言的基本数据类型有：

```
char 字符型, int 整数型  
float 浮点数型 (又称为单精度数)  
double 双精度浮点数型
```

上述数据类型的长度及存储的值域如表 2.1 所示。

表 2.1 数据类型的长度及存储的值域

类 型	占字节长度(位)	值 域
char	8	0~255
int	16	-32768~32767
float	32	3.4E-38~3.4E+38
double	64	1.7E-308~1.7E+308

此外, 还有一些用于整型的限定词: short (短)、long (长)、signed (有符号) 和 unsigned (无符号), short 和 long 表示不同长度的整型量; unsigned 表示无符号整型, 它的存放值总是正的; signed 可以省略。

例如: short int x;
long int y;
unsigned int z;

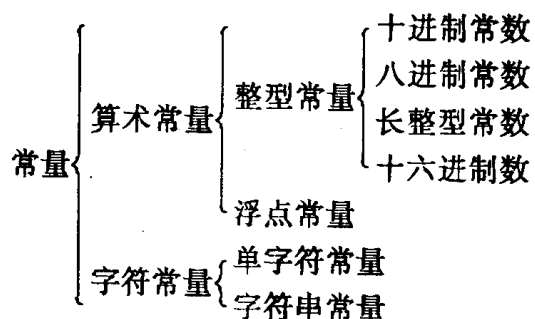
在这种情况下, 可以把说明符 int 省略, 加了限定词的数据类型及它们的长度和取值范围如表 2.2 所示。

表 2.2 加限定词的数据类型及其长度和取值范围

类	型	占字节长度(位)	值域
	short int	16	-32768~32767
	long int	32	-2147483648~2147483647
	unsigned int	16	0~65535

2.3 常量

C 语言中的常数可分为算术常量和字符常量二部分, 它们有如下类型:



2.3.1 整型常量

C 语言中的整型常量可以用四种形式表示, 它们是十进制常数、长整型常数、八进制常数和十六进制常数。

十进制常数一般占一个机器字长, 是一个带正负号的常数, 所以该常数是一个有限的整数集合, 它的取值范围在 $-2^{(n-1)} \sim 2^{(n-1)}-1$ 之间, n 为机器字的位数, 如 IBM-PC 系列机数的范围为 $-32768 \sim 32767$ 。下列常数都是合法的十进制整型常数:

-32768 , 0 , 32767 , $+123$, -456 , 987

长整型常数与十进常数一样, 差别仅在于长整型常数一般占两个机器字长, 其数值范围为 $-2^{(2n-1)} \sim 2^{(2n-1)}-1$ 。为了区分, 规定长整型常数是在十进制数后加上字母 L (或 l)。如下