

杨 磊 主 编
胡局新 副主编

C语言程序设计

中国矿业大学出版社

China University of Mining and Technology Press

C 语言程序设计

主 编 杨 磊

副主编 胡局新

参 编 吴 川 吴雪峰

赵 芳 王 爽

中国矿业大学出版社

内 容 提 要

本书较全面地讲述了 C 语言程序设计的基础知识,括基本数据类型和运算符、控制结构、函数、数组、指针、结构与共用、文件以及程序设计实例。每一章都附有精选的、多种类型的练习题,有助于读者复习、巩固所学知识,培养读者的实际编程能力。本书可以作为各类高等院校、高职院校计算机专业及理工科非计算机专业学生学习“计算机程序设计”课程的教材,也可作为广大计算机爱好者学习 C 程序设计语言的参考书。

图书在版编目(CIP)数据

C 语言程序设计:含实验 / 杨磊主编. —徐州:
中国矿业大学出版社, 2011. 8
ISBN 978 - 7 - 5646 - 1226 - 9
I. ①C… II. ①杨… III. ①C 语言—程序设计 IV.
①TP312
中国版本图书馆 CIP 数据核字(2011)第 181338 号

书 名 C 语言程序设计
主 编 杨 磊
责任编辑 仓小金
出版发行 中国矿业大学出版社有限责任公司
(江苏省徐州市解放南路 邮编 221008)
营销热线 (0516)83885307 83884995
出版服务 (0516)83885767 83884920
网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com
印 刷 徐州中矿大印发科技有限公司
开 本 787×960 1/16 本册印张 16 本册字数 305 千字
版次印次 2011 年 8 月第 1 版 2011 年 8 月第 1 次印刷
总 定 价 40.00 元(共两册)
(图书出现印装质量问题,本社负责调换)

前 言

《C 语言程序设计》是为高等院校 C 语言程序设计课程而编写的教材。全书分为十二章,主要内容包括:C 语言概述、数据类型合表达式、算法、顺序结构、选择结构、循环结构、数组、指针、函数和作用域、复杂数据类型、位运算、文件的操作。本教材在结构上突出了以程序设计为中心、以语言知识为工具的思想,对 C 语言的语法规则进行了整理和提炼,深入浅出地介绍了它们在程序设计中的应用;在内容上注重知识的完整性,适合初学者的需要;在写法上追求循序渐进,内容由浅入深,示例丰富充实,重点突出,通俗易懂。

在本书的编写过程中,以实用、够用、会用为主要编写目的,在内容的选取上,强调掌握 C 语言基础知识,重点掌握编程的方法和技巧,在每一章节的编写中都配备相关的范例,使读者能经过实际操作后真正掌握所学的知识要点。在每章之前都要有一个课前预习案例,方便学生通过实例尽快地了解本章的主要内容,并且每章均提供了大量的习题。

本书既可以作为高等院校非计算机专业学生的计算机语言教材,也可以作为高等院校计算机专业本科、专科低年级学生学习计算机语言的入门教材。也可作为全国计算机等级考试(二级)及各种培训班的教材,同时也可作为广大计算机爱好者的入门参考书。

本书中的所有参考程序都由杨磊、吴雪峰、吴川、王爽、赵芳等老师上机调试通过。由于作者水平有限,本书在编写过程中难免有疏漏之处,敬请广大读者提出宝贵意见,以便改进。作者的 E-mail 地址为:xuhaiyl@163.com.。

编 者

2011 年 7 月

目 录

第一章 C语言概述	1
第一节 C语言的发展史	1
第二节 C语言的特点	3
第三节 C程序的结构特点	5
第四节 C语言的字符集和词汇	10
第五节 C语言的运行步骤与方法	12
第六节 同步练习	18
第二章 数据类型和表达式	19
第一节 基本数据类型	20
第二节 常量与变量	22
第三节 整型数据	25
第四节 实型数据	27
第五节 字符型数据	29
第六节 数据类型转换	32
第七节 算术运算符与算术表达式	34
第八节 赋值运算符与赋值表达式	37
第九节 逗号运算符与逗号表达式	39
第十节 本章小结	41
第十一节 同步练习	42
第三章 程序的灵魂——算法	45
第一节 算法的概念	45
第二节 算法的特性	46
第三节 怎样表示一个算法	47
第四节 算法的基本逻辑结构	48
第五节 基本算法举例	51

第六节 同步练习	53
第四章 顺序结构	54
第一节 顺序结构概述	54
第二节 C 语言数据输入输出的实现	56
第三节 格式输入输出函数	57
第四节 字符的输入输出	67
第五节 本章小结	69
第六节 同步练习	69
第五章 选择结构	71
第一节 选择结构程序设计	71
第二节 关系运算符与关系表达式	72
第三节 逻辑运算符与逻辑表达式	73
第四节 if 语句	78
第五节 条件运算符与条件表达式	85
第六节 switch 语句	87
第七节 本章小结	89
第八章 同步练习	89
第六章 循环结构	91
第一节 循环结构	92
第二节 goto 语句	92
第三节 while 语句	93
第四节 do while 语句	94
第五节 for 语句	97
第六节 嵌套循环	100
第七节 break 与 continue 语句	100
第八节 本章小结	103
第九节 同步练习	103
第七章 数据仓库——数组	104
第一节 一维数组的定义和引用	105
第二节 二维数组的定义和引用	110

第三节	字符数组	114
第四节	程序举例	120
第五节	本章小结	124
第六节	同步练习	124
第八章	指针	128
第一节	指针概述	129
第二节	数组与指针	134
第三节	字符串与指针	149
第四节	函数与指针	156
第五节	指针数组和指向指针的指针	160
第六节	本章小结	167
第七节	同步练习	168
第九章	函数和作用域	170
第一节	函数的定义	171
第二节	函数的参数传递	176
第三节	递归函数	182
第四节	作用域	186
第五节	库函数	190
第六节	本章小结	191
第七节	同步练习	192
第十章	复杂数据类型	194
第一节	结构体	195
第二节	枚举类型	199
第三节	一个经典实例:栈结构	200
第四节	本章小结	204
第五节	同步练习	205
第十一章	位运算	206
第一节	位的运算	206
第二节	本章小结	214
第三节	同步练习	215

第十二章 文件的操作	216
第一节 文件的概念.....	217
第二节 文件类型指针.....	218
第三节 文件的打开与关闭.....	219
第四节 文件的读写.....	222
第五节 文件的定位.....	234
第六节 出错的检测.....	237
第七节 文件输入输出小结.....	237
第八节 同步练习.....	238
参考文献	244

第一章 C语言概述

在学习 C 语言之前,首先要了解 C 语言的发展历程,这是每一个刚刚学习 C 语言的人员都应该清楚的,并且要了解为什么要选择 C 语言,以及它有哪些特性。只有了解了 C 语言的历史和特性,才会更为深刻地了解这门语言,并且增加对今后学习 C 语言的信心。随着计算机科学的不断发展,C 语言的学习环境也在不断变化,刚开始学习 C 语言时,大多数人会选择一些相对简单的编译器,如 Turbo C 2.0。但是,现在更多的人还是选择了由 Microsoft 公司推出的 Visual C++ 6.0 编译器。

本章致力于使读者了解 Visual C++ 6.0 的开发环境,掌握 Visual C++ 6.0 集成开发环境中各个部分的使用方法,并能编写一个简单的应用程序。

课程目标

- ◇ 了解 C 语言出现的历史背景
- ◇ 了解 C 语言的特点
- ◇ 掌握 C 语言的基本结构
- ◇ 掌握如何使用 Visual C++ 6.0 开发 C 程序

第一节 C语言的发展史

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。C 语言对操作系统和系统使用程序以及需要对硬件进行操作的场合,用 C 语言明显优于其他高级语言,许多大型应用软件都是用 C 语言编写的。C 语言具有绘图能力强、可移植性高并具备很强的数据处理能力,因此适于编写系统软件,三维、二维图形和动画,它是数值计算的高级语言。

一、计算机语言的发展过程

在介绍 C 语言的发展历程之前,先来对程序语言进行大概了解。

1. 机器语言

机器语言是低级语言,也称为二进制代码语言。计算机使用由“0”和“1”二

进制数组成的一串串指令来进行工作的。机器语言的特点是计算机可以直接识别,不需要进行任何的翻译。

2. 汇编语言

汇编语言是面向机器的程序设计语言。为了降低使用机器语言编程的复杂程度,人们用英文字母或符号串来替代机器语言的二进制码,即汇编语言。汇编语言比机器语言便于阅读和理解。

3. 高级语言

由于汇编语言依赖于硬件体系,并且该语言中的助记符号数量比较多,所以其使用起来仍然不够方便。为了使程序语言能更贴近人类的自然语言,同时又不依赖于计算机硬件,于是发明了高级语言。这种语言,其语法形式类似于英文,并且不直接操作硬件,易于被普通人所理解与使用。其中影响较大、使用普遍的高级语言有 Fortran、Algol、Basic、Cobol、Lisp、Pascal、Prolog、C、C++、VC、VB、Delphi、Java 等。

我们可以通过图 1-1 直观地看出计算机语言的发展流程。

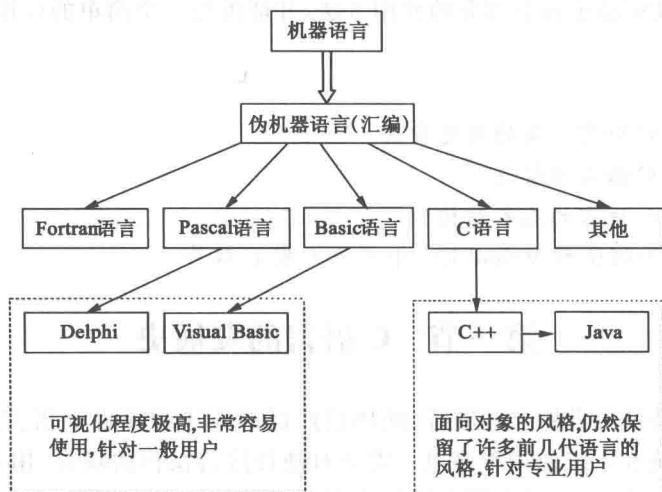


图 1-1 计算机语言发展流程图

二、C 语言的发展历程

C 语言的发展颇为有趣,它的原型 Algol 60 语言。1963 年,剑桥大学将 Algol 60 语言发展成为 CPL(combined programming language)语言。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。

1970 年,美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改,并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干,提炼出它的精华。并且他用 B 语言写了第一个 UNIX 操作系统。而在 1973 年,B 语言也被“煮”了一下,美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言,他取了 BGPL 的第二个字母作为这种语言的名字,这就是 C 语言。

为了使 UNIX 操作系统得以推广,1977 年 Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年 Brian W. Kernighan 和 Dennis M. Ritchie 出版了名著 *The C Programming Language*,从而使 C 语言成为目前世界上流行最广泛的高级程序设计语言。

1988 年,随着微型计算机的日益普及,出现了许多 C 语言版本。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准协会(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。C 语言发展迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件,如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序,就更能显示 C 语言的优势了,像 DOS、WORDSTAR 等就是用这种方式编写的。

第二节 C 语言的特点

一、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。它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。C 语言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,共有种 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。因此 C 语言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,且计

算功能、逻辑判断功能强大。

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 语言是以函数形式提供给用户的,这些函数可方便地调用,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,能够检查出几乎所有的语法错误。而 C 语言允许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

因此,C 语言既具有高级语言的功能,又具有低级语言的特征,能够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,可以用来写系统软件。

7. 程序生成代码质量高,程序执行效率高

一般只比汇编程序生成的目标代码效率低 10%~20%。

8. 适用范围大,可移植性好

C 语言有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,也适用于多种机型。

如果读者在初次接触 C 语言则有可能感觉不到 C 的各种优点,这需要再进一步学习后进行感受,一句话“谁用谁知道”。

二、C 语言的应用

C 语言一般用来进行底层开发,如操作系统、嵌入式系统的开发,或者用在要求高效率、高可移植性的地方。C 语言对人要求很高,程序员要考虑的地方太多,它的特点就是每一个字节都可以精确控制,不像 JAVA、C# 等,编译器自动加的东西太多,效率也就低了。Windows 就是用 C 语言来开发的,LINUX 很多程序也用 C 语言编写。

当今流行的编程软件种类繁多,它们编程方便、易于维护,但是在与硬件直接打交道和编制系统软件时却束手无策,于是 C 语言就有了用武之地。C 语言作为汇编语言与高级语言之间的一种过渡语言,兼有汇编语言的高效性和高级语言的方便性。

当然了 C 语言除了底层的开发之外,很多的游戏、应用工具、用户系统都可以由 C 语言开发。

第三节 C 程序的结构特点

进行 C 语言程序学习之前,首先不要对 C 语言产生恐惧感,觉得这种语言都应该是学者或研究人员才能够掌握的。C 语言是人类共有的财富,只要通过努力学习就可以掌握。下面通过三个简单的程序来看一看 C 语言程序是什么样子。

一、简单 C 语言程序介绍

【例 1.1】 输出 hello C。

```
#include<stdio.h>

void main( )
{
    printf("hello C!\n");
}
```

本程序的作用是输出以下一行信息:

hello C!

现在来分析一下上面的实例程序。

(1) 第一行: #include 指令

实例代码中的第一行: #include<stdio.h>, 这个语句的功能是进行有关的预处理操作。include 称为文件包含命令;后面尖括号中的内容,称之为头部文件或首文件,它的作用是让用户下面使用 printf() 函数时合法。

(2) 第二行: main() 函数

main 是主函数的函数名(函数头),表示这是一个主函数。每一个 C 源程序都必须有 main 函数且只能有一个主函数(main 函数)。main 函数就是一个程序的入口部分。也就是说,程序都是从 main 函数头开始执行的,然后进入到 main 函数中,执行 main 函数中的内容。main 前面的 void 表明此函数是“空类型”,即此函数被执行后不产生一个函数值(但是有的函数执行后是有返回值的比如: sin(x))。

(3) 第三行和第五行: { 函数体 }

程序代码中的第 3 行和第 5 行这两个大括号就构成了函数体,函数体也可以称为函数的语句块。在函数体中,也就是第 4 行这部分就是函数体中要执行的内容。{ } 一定是成对出现的,可以说是成双成对,有始有终。

(4) 第四行: 执行语句

执行语句就是函数体中要执行的动作内容。这一行代码是这个简单的例子

中最复杂的。该行代码虽然看似复杂,其实也不难理解,printf 是产生格式化输出的函数,可以简单理解为向控制台进行输出文字或符号。在括号中的内容称为函数的参数,括号内可以看到输出的字符串"hello C"。其中可以看到\n 这样一个符号,称之为转义字符。转义字符的内容会在后面章节进行介绍。

(5) 空格

我们发现在写 printf 函数的时候并没有顶格写,这是为什么呢?

C 语言是一个较灵活的语言,因此格式并不是固定不变、拘于一格的。也就是说空格、空行、跳格并不会影响程序。这就像生活中在纸上写字一样,虽然拿来一张白纸就可以在上面写字,但是通常还会在纸的上面印上一行一行的方格或段落,隔开每一段文字,自然就美观和规范。合理、恰当地使用这些空格、空行,可以使编写出来的程序更加规范,对日后的阅读和整理也很重要。在此也提醒读者,在写程序的时候最好将程序书写得规范、干净。

【例 1.2】 求两个数之和。

/* 以下程序求两个数 a、b 的和 sum 值,这是我们第二个程序,希望大家尽快进入状态 */

```
#include<stdio.h>

void main()                                //求两个数之和
{
    int a,b,sum;                          //变量的声明部分,定义 a、b、sum 为整型
    a=1;b=2;                              //为变量 a、b 赋值
    sum=a+b;                              //求 a+b 的结果,并赋值给 sum
    printf("sum=%d\n",sum);               //打印出 sum 值
}
```

本程序的作用是求两个整数 a、b 之和 sum。

(1) 注释

“/* */”与“//”所起的作用都是注释,目的是为了程序更具有可读性,注释对编译和运行不起作用。通常/* */是多行注释,//是单行注释。

(2) 变量声明

第四行是声明部分,定义整型(int)变量 a、b 和 sum。在 C 语言中要使用变量,必须在使用变量之前进行定义,之后编译器会根据变量的类型为变量分配内存空间。变量的作用就是存储数值,用变量进行计算。这就像在二元一次方程中,X 和 Y 就是变量,当为其进行赋值后,例如 X 为 5,Y 为 10,这样 X+Y 的结果就等于 15。

(3) 赋值语句

第五行、第六行是赋值语句,使 a 和 b 的值分别为 1 和 2。第六行求 sum 的值。

(4) 格式字符串

第七行中的“%d”是输入输出的“格式字符串”,“%d”表示“以十进制整数类型”,用来指定输入输出时的数据类型和格式。这可以看出 C 语言在输出格式上是比较严谨的。编译器在执行 printf 语句时会自动进行判断哪些是直接输出的,哪些是有特殊含义的,比如在第七行中,“sum=”直接打印输出,有“%d”命令编译器就知道是格式字符,有“\n”命令编译器就知道是转义符,而 printf 函数中括号内逗号的右端 sum 是要输出的变量,它的值为 3,3 的位置应该出现在“%d”的位置上。

所以此程序的运行结果为:

```
sum=3
```

【例 1.3】 求长方体的体积。

本实例要实现这样的功能:有一个长方体,它的高已经给出,然后输入这个长方体的长和宽,通过输入的长、宽以及给定的高度计算出这个长方体的体积。

```
#include<stdio.h>                                /*包含头文件*/
#define Height 10                                  /*定义常量*/
int calculate(int Long, int Width);                /*函数声明*/
int main()                                          /*主函数 main*/
{
    int m_Long;                                    /*定义整型变量,表示长度*/
    int m_Width;                                    /*定义整型变量,表示宽度*/
    int result;                                     /*定义整型变量,表示长方体的体积*/
    printf("长方形的高度为:%d\n",Height);          /*显示提示*/
    printf("请输入长度\n");                          /*显示提示*/
    scanf("%d",&m_Long);                             /*输入长方体的长度*/
    printf("请输入宽度\n");                          /*显示提示*/
    scanf("%d",&m_Width);                             /*输入长方体的宽度*/
    result=calculate(m_Long,m_Width);                /*调用函数,计算体积*/
    printf("长方体的体积是:");                      /*显示提示*/
    printf("%d\n",result);                          /*输出体积大小*/
    return 0;                                        /*返回整型 0*/
}
int calculate(int Long,int Width)                  /*定义计算体积函数*/
```

```

int result=Long*Width*Height;           /*具体计算体积*/
return result;                           /*将计算的体积结果返回*/
}

```

运行结果如下:

长方形的最高度为:10

请输入长度

5

请输入宽度

4

长方体的体积是:200

Press any key to comtinue_

以下介绍程序中的关键元素。

(1) 定义常量

实例代码中的第 2 行:

```
#define Height 10           /*定义常量*/
```

这一行代码中,使用 #define 定义一个符号,#define 在这里的功能是设定这个符号为 Height,并且指定这个符号 Height 代表的值为 10。在程序中,只要是使用 Height 这个标识符的位置,就代表使用的是 10 这个数值。

(2) 函数声明

实例代码中的第 3 行:

```
int calculate(int Long,int Width);    /*函数声明*/
```

此处代码的作用是对一个函数进行声明,前面介绍过函数的声明,但是什么是声明函数呢? 举一个例子,两个公司进行合作,其中的 A 公司要派一个经理到 B 公司进行业务洽谈。A 公司会发送一个通知给 B 公司,告诉 B 公司会派一个经理过去,请 B 公司在机场接一下这位洽谈业务的经理。A 公司将这位经理的名字和大概的体貌特征都告诉 B 公司的有关迎接人员。这样当这位经理下飞机之后,B 公司就可以将他的名字写在纸上作成接机牌,然后找到这位经理。

声明函数的作用就像 A 公司告诉 B 公司有关这位经理信息的过程,为接下来要使用的函数作准备。也就是说如果此处声明 calculate 函数,那么在程序代码的后面会有 calculate 函数的具体定义内容,这样程序中如果出现 calculate 函数,程序就会根据 calculate 函数的定义执行有关的操作。至于有关的具体内容将会在第九章中进行介绍。

(3) 输入语句

实例代码中第 15 行:

```
scanf("%d",&m_Long); /*输入长方体的长度*/
```

在前面的实例中,曾经介绍过显示输出函数 printf,那么既然有输出就一定会有输入。在 C 语言中,scanf 函数就用来接受键盘输入的内容,并将输入的结构保存在相应的变量中。可以看到,在 scanf 的参数中,m_Long 就是之前定义的整型变量,它的作用是存储输入的信息内容。其中的 & 符号是取地址运算符,其具体内容将会在本书的后续章节中进行介绍。

(4) 调用函数

通过主函数调用 calculate 函数,在调用时将实际参数 m_Long 和 m_Width 的值分别传给 calculate 函数中的参数 Long 和 Width(形式参数)。经过执行 calculate 函数得到一个返回值(即 calculate 函数中的 result),这个值返回到调用 calculate 函数的位置,即程序第 14 行“=”的右侧,然后把这个值赋给变量 result(注意此 result 与 calculate 函数中的 result 不同)。

本例中所涉及的函数调用、参数传递、返回值的概念,在此只是作了简单的解释,读者可以不予深究,在后续有关章节的学习中自会迎刃而解。

二、C 语言程序的格式

通过上面的两个实例的介绍可以看出 C 语言编写格式有以下特点。

1. 主函数 main

一个 C 程序都是从 main 函数开始执行的。main 函数可以放在任何位置。

2. C 程序整体是由函数构成的

程序中 main 就是其中的主函数,当然在程序中是可以定义其他函数的。在这些定义函数中进行特殊的操作,使得函数完成特定的功能。虽然将所有的执行代码全部放入 main 函数也是可行的,但是如果将其分成一块一块,每一块使用一个函数进行表示,那么整个程序看起来就具有结构性,并且易于观察和修改。

3. 函数体的内容在“{}”中

每一个函数都要执行特定的功能,那么如何才能看出一个函数的具体操作的范围呢?答案就是寻找“{”和“}”这两个大括号。C 语言使用一对大括号来表示程序的结构层次,需要注意的就是左右大括号要对应使用。

4. 每一个执行语句都以“;”结尾

如果注意观察前面的两个实例,就会发现在每一个执行语句后面都会有一个“;”(分号)作为语句结束的标志。

5. 英文字符大小通用

在程序中,可以使用英文的大写字母,也可以使用英文的小写字母。但一般