C程序设计语言题解与示例

宋秋杰　王金汉　编著

429
C5

同济大学出版社

838498-507
73.87429
055-C5

C程序设计语言题解与示例

宋秋杰
王金汉
编著

同济大学出版社

内 容 提 要

本书是《C程序设计语言》一书的续篇。全书对《C程序设计语言》一书的各章分别给出相应的题解及示例，对其中较复杂的函数和较大的程序在计算机上进行过验证，对示例中较难理解的部分作了扼要说明。阅读和操练这些题解和示例有助于读者熟练地掌握C语言的编程技巧及进一步学习UNIX操作系统。

本书可作为高等院校计算机专业的辅助教材，也可供从事计算机工作的技术人员参考。

责任编辑　王兴伟
　　张　颖
封面设计　王肖生

C程序设计语言题解与示例

宋秋杰　王金汉编著

同济大学出版社出版

（上海四平路1239号）

新华书店上海发行所发行

复旦大学印刷厂印刷

开本：787×1092　1/16　　印张：7.25　　字数：181千

1989年11月第1版　　1989年11月第1次印刷

印数：1—3100　　定价：4.3元

ISBN 7-5608-0426-9/TP·26

序　言

《C 程序设计语言》（以下简称《C 语言》）一书出版发行后，受到了专业技术人员和高校师生欢迎。本书是 C 语言习题的解答，目的是让读者自我检验学习效果。为了进一步提高读者编制程序的水平，书中还编制了大量的示例。

本书编写结构是：对应《C 语言》一书各章分别编写相应题解及示例，对其中较大的函数和程序在计算机上进行了验证。读者可与自己编制的题解对照参考。

对本书给出的题解及示例有三点需说明：第一，每个函数不是直接就可投入运行的程序。例如，它们都没有通过#*include* 〈*stdio.h*〉将标准库文件 *stdio.h* 包括在源文件中。第二，不是最佳或最完备的。例如，第二章第 9 题的题解中，没有对作为输入数据的三角形三条边长 *a*,*b*,*c* 值的正确与否进行判断，如要在计算机上运行，必须对输入的 *a*,*b*,*c* 进行校验。第三，题解可能有多种，本书仅给出一种或两种。读者编制的程序可能比题解更简洁易懂。

为了使读者更好地掌握 C 语言及编程技巧，并能实际应用，在各章之后有选择地编制了一些示例。示例中对较难理解的部分作了简要说明，有助于读者阅读。特别要指出，大部分示例在程序设计技巧上是完全可以效法的。

在编写本书过程中，得到了上海市计算技术研究所黄兴同志的热忱帮助，他在验证题解等方面给予作者很大的帮助，在此表示感谢。鉴于作者学识水平有限，加上编写较匆促，可能有不足之处，请读者批评指正。

作者

1988年1月

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第一章 引 论

习 题 一

1．简述C语言的主要特征和应用领域的适用性。

题解：

C语言由于它具有描述问题能力强，灵活性好，应用面宽，编译实现容易，目标质量高，语言规模和编译程序规模相当小，通用性和可移植性良好等特点，使C语言深受系统软件和应用软件的设计工作者欢迎。C语言主要特征：

(1) 以函数为基本单位的模块化程序结构，便于实现结构程序。

(2) 有利于结构化的程序控制流语句群，方便实施。

(3) 多种数据类型定义，利于数据结构化。

(4) 以单一的值参数传递方式，简单明了地实现函数间的通讯。

(5) 以指针统一了指针、地址计算、数组和结构等数据类型。

(6) 具有宏定义预处理手段，有利于实现程序的完整性及库函数的引用。

(7) 跨函数跨源程序文件的变量存贮类型，统一实现变量公用化。

(8) 具有34种运算和15种优先级运算顺序，由此可产生经济、实用、灵活的表达式。

(9) 一个完整程序可由一个或多个源程序文件组成，便于结构化程序的实现。

(10) 简洁的程序表示和习惯的书写格式。

由于C语言与UNIX操作系统联系紧密，随着UNIX的深入使用，C语言也得到广泛推广应用。C语言虽然不象PASCAL语言有强类型数据定义，但它有实现容易，目标质量高等特点，成为系统软件和应用软件各个邻域的有力工具。具体地说，它的适用性与应用环境有关。

C语言是商用数据处理和科学数值计算的可适用工具，但不是最佳或最理想的工具。

C语言由于它的灵活性和对硬件寄存器的直接描述，是理想的系统程序设计工具。它也易于实现系统实用程序（例如，数据库、语言编译、编辑等）。

根据具体应用环境，决定选用的语言。如以安全、可靠为主要设计目标的，可选用PASCAL。如以描述力强、灵活性高、容易实现为主要设计目标的，可选用C语言。当然要实现某一软件，多数是采取多种软件工具，联合使用。

2．现代良好的程序设计风格是什么？在开发一个具体软件和软件工程项目时，如何体现？

题解：

设计一个软件时，其设计风格可能随着应用目标不同而略有差异。但在现代计算机背

景下，程序设计的良好风格有一定程度的统一性，其主要方面是：

(1) 自顶向下的系统工程的开发顺序，把一个大的项目逐级分解成多层次的模块结构。易实现易修改。

(2) 程序控制流结构化，易读易理解。

(3) 数据结构系统化，这样可使算法和数据结构的设计既分别展开，又有联系地进行。

C 语言之所以广泛应用，一个主要原因是它与现代良好的程序设计风格相匹配。

3. 按你的理解和实践，说明程序设计质量标准有哪些要素？试按重要性列出。

题解：

在现代的计算机上研制软件并不只是为了求得一个正确的程序，而要求具备下列要素：

(1) 是一个正确的软件；

(2) 符合现代良好的程序设计风格；

(3) 易编、易读、易修改；

(4) 可移植性好，通用性强；

(5) 占用系统资源少，运行效率高。

4,5,6题略。

第二章　数据、表达式及输入输出简介

习　题　二

1．C语言中的标识符是怎样构成的？下列符号中哪些是正确的标识符，哪些是错误的标识符，为什么？

a5b　5h4　pel.1　g1(x)　case

s_i_1　$300　e-10　xyz　next　day

题解：

标识符是以字母开头的字母与数字的序列。连字符（低短线_）也看作为字母。

正确的标识符是：

a5b　s_i_1　xyz

错误的标识符是：

5h4　非字母开头。

pel.1　出现非字母和数字符号.。

g1(x)　出现非字母和数字符号（和）。

case　与保留字同名。

$300　非字母开头。

e-10　出现非字母和数字符号-。

next day　在中间出现空格字符。

2．下列常数中哪些是合法的常数，哪些是非法的常数？对于合法的常数，指出所属类型；对于非法的常数，说明错误原因。

-78　+38.40　16,000　-0.　01377　0x2a1　0x3g　e13

3.e-8　'\0'　"x"　'{'　'101'　'\55'　"101"　"36'56""

题解：

合法常数是：

-78	01377	0x2a1	整型
+38.40	-0.	3.e-8	实型
'\0'	'{'	'\55'	字符型
"x"	"101"		字符串

非法的常数是：

16,000　逗号是非法的。

0x3g　十六进制整数中不能出现字母g。

e13　实型常数中既无整数部分，又无小数部分。

'101'　在两个撇号中出现了三个字符。

"36′56″″ 最后的双撇号是多余的。

3. 下面是一串语句，请指出执行每个语句后整型变量 k,m,n 的值。

```
k=0;
n=k++;
m=(k*=2);
k=--n;
n+=(n++)-n;
```

题解：

语　　句	k	m	n
k=0;	0	未定值	未定值
n=k++;	1	未定值	0
m=(k*=2);	2	2	0
k=--n;	−1	2	−1
n+=(n++)−n;	−1	2	−1

4. 修改例 2-1 中的函数，使其功能是将第一个整数的高字节作为结果的高字节，第二个整数的高字节作为结果的低字节。

题解：

```
combytes(k,p)
unsigned k,p;
{
  return((k & 0177400)|(p & 0177400)≫8);
}
```

5. 修改例 2-2 的要求：假定最左边的一位是第 0 位，从左向右编号。

题解：

```
getbits(x,p,n)
unsigned x,p,n;
{
  return((x≪p)&~(~0≫n));
}
```

6. 若 i 为整型变量，x 为浮点变量，将下列表达式中所隐含的类型的自动转换用强制运算符显式地表示出来：

```
(i+'0')-x
('2'-'0')+i*x
```

```
x=(i<3.0)?-1.0:1
```

题解：

```
(float)(i+(int)'0')-x
(float)((int)'2'-(int)'0')+(float)i*x
x=((float)i<3.0)?-1.0:(float)1
```

7．写一个函数 lower，将读入的字母从大写转化成小写后输出，非大写字母按原样输出。

题解：

```
lower( )
{
  int c;
  c=getchar( );
  return(c>='A'&& c<='Z'?putchar(c+'a'-'A'):putchar(c));
}
```

8．写一个函数，将读入的数字转化成相应的整数值输出。如将数字'8'转化成数值8，然后输出。

题解：

```
transdigit( )
{
  printf("%d",getchar( )-'0');
}
```

9．已知三角形三条边 a,b,c，求三角形面积的公式为：

$$f=\sqrt{s(s-a)(s-b)(s-c)}$$

$$s=(a+b+c)/2$$

编一个程序，读入 a,b,c 的值，计算三角形的面积 area，输出三条边长及面积的值。

题解：

```
main( )
{
  float a,b,c,s;
  printf("please enter the sides here:");
  scanf("%f%f%f\n",&a,&b,&c);
  s=(a+b+c)/2;
  printf("a=%fb=%fc=%farea=%f",a,b,c,sqrt(s*(s-a)*(s-b)*(s-c)));
}
```

10．从终端打入两个字符给变量 c1 和 c2，将较大者放到 c1，较小者放到c2，并输出较大者。

题解：

```
maxc( )
{
  char c,c1,c2;
  (c1=getchar( ))>(c2=getchar( ))?(c=c1):(c=c2,c2=c1,c1=c);
  putchar(c);
}
```

示　例

2-1.在你的系统中运行下列程序，考察得到什么样的结果和出错信息。

```
main( )
{
  printf("hello,world");
}
```

此例中无换行符（\n），故输出后不换行。

```
main( )
{
  printf("hello,world\n")
}
```

此例中 printf( )后没有分号。由于每个 C 语句必须由分号结束，故编译程序应发现此错误，并打印相应的出错信息。

```
main( )
{
  printf("hello,world\n');
}
```

本例的 \n 之后是一个单撇号而不是双撇号，故单撇号连同右括号及分号都被认为是字符串的组成部分。编译程序将发现这个错误，输出的出错信息是：漏双撇号或字符串太长。

2-2.所给出的程序的三个 printf 的输出字符串中都含有 \x 的形式，其中 x 是在《C 程序设计语言》一书的第19页上未列出的某个字符。运行这个程序，考察输出的结果是什么。

```
main( )
{
  printf("hello,world\a");
  printf("hello,world\7");
  printf("hello,world\?");
}
```


此为试读，需要完整PDF请访问：www.ertongbook.com

运行此程序所输出的结果取决于编译程序，有的实现方案中将反斜线略去，有的方案中把这种 \x 视为无定义。故一种可能的输出结果是：

hello,worldahello,world〈BELL〉hello,world?

其中〈BELL〉表示由 ASCII 码值 7 所代表的短促的嘟嘟声。

2-3. 编一个函数 invert(x, p,n)，其功能是翻转 x 中第 p 位开始的 n 个二进 位，x 中的其它二进位的内容不变。翻转的意思是将 1 变 0，0 变 1。

```
invert(x,p,n)/* inverts n bits of x beginning at position P */
unsigned x, p, n;
{
  return(x∧(~(~0≪n)≪p));
}
```

首先执行(~0≪n)，将全 1 左移 n 位，使最右边的 n 位为 0。接着是~(~0≪n)，使最右边的 n 位变为 1，其余位为 0。

然后执行(~(~0≪n)≪p)，将最右边的 n 位 1 移到从 p 位开始的 n 位。

最后执行x∧(~(~0≪n)≪p)。由于按位加∧操作的结果是：当两个二进位不同时得 1，否则为 0。故执行的结果使 x 中从 p 位开始的 n 位中的内容由原 来 的 1 变 0，原 来 的 0 变 1；而这 n 位以外的其它位保持不变，这就达到了翻转的目的。

第三章　流程控制和复合语句

习　题　三

1．修改例 3-2 的程序，当方程有两个不同的实根时，先计算绝对值大的那个根 x1，然后用公式 x2＝c/(a＊x1) 计算绝对值较小的那个根。

题解：

```
main( )
{
  double a,b,c,disc,twoa,term1,term2;
  printf("enter a,b,c:");
  scanf("%f%f%f",&a,&b,&c);
  if(a--0)
  if(b--0)
    printf("no answer due to input error\n");
  else
    printf("the single root is %f\n",-c/b);
  else
{
  disc=b*b-4*a*c;
  twoa=2*a;
    term1=-b/twoa;
    term2=sqrt(abs(disc))/twoa;
    if(disc<0)
      printf("complex roots\n real part=%f image part=%f\n", term1,term2);
    else
    { if(term1*term2>0)term1=term1+term2;
      else term1=term1-term2;
      printf("real roots\n root1=%f root2=%f\n",
            term1,c/(a*term1));
    }
  }
}
```

2．若有如下条件语句：

```
if(a<b)if(c<d)x=1;
```

```
else if(a<c)if(b<d)x=2;
                  else x=3;
else if(a<d)if(b<c)x=4;
                  else x=5;
                  else x=6;
                  else x=7;
```

(a) 根据 if 和 else 配对的原则，用对齐格式重写此语句。

(b) 是否有多余的判别条件或矛盾的判别条件？

(c) 重写一个与此条件语句等效的更简单的条件语句。

题解：

(a)
```
if(a<b)
  if(c<d)x=1;
  else
    if(a<c)
      if(b<d)x=2;
      else x=3;
    else
      if(a<d)
        if(b<c)x=4;
        else x=5;
      else x=6;
else x=7;
```

(b) 当 a<b 且 c>=d, a>=c 时，a<d 是不可能的，故此条件语句中永远不可能执行 x=4; 和 x=5; 两个语句。当 a<b 且 c>=d, a>=c 时，a 肯定大于或等于 d，故此时再判别 a>=d 是多余的。

(c) 根据(b)中分析，可将原条件语句等效地写成：

```
if(a<b)
  if(c<d)x=1;
  else
    if(a<c)
      if(b<d)x=2;
      else x=3,
    else x=6;
else x=7;
```

3. 写一个程序，计算输入正文中的空格数、制表符数和换行符数。

题解：

```
main( )/* count blanks, tabs, and new lines */
{
```

```
  int c, nb, nt, nl;
  nb=nt=nl=0;
  while((c=getchar( ))!=EOF)
  { if(c--' ')++nb;
    if(c--'\t')++nt;
    if(c--'\n')++nl;
  }
  printf("%d %d %d\n",nb,nt,nl);
}
```

4. 写一个程序，将输入复制到输出，当输入正文中有连续一个以上的空格符时，在输出中用一个空格符代替。

题解：

```
#define NONBLANK 'a'
main( )  /* replace string of blanks with a single blank */
{
  int c, lastc;
  lastc=NONBLANK;
  while ((c=getchar( ))!=EOF)
  { if(c!=' ')putchar(c);
    if(c--' ')
      if(lastc !=' ')putchar(c).
    lastc=c;
  }
}
```

也可写成：

```
#define NONBLANK 'a'
main( ) /* replace string of blanks with a single blank */
{
  int c, lastc;
  lastc=NONBLANK;
  while((c= getchar( ))!=EOF)
  { if(c!=' ')putchar(c);
    else if(lastc !=' ')
          putchar(c);
    lastc=c;
  }
}
```

还可写成：

```
#define NONBLANK 'a'
main( )/* replace string of blanks with a single blank */
{
  int c,lastc;
  lastc=NONBLANK;
  while((c=getchar( ))!=EOF)
  { if(c!=' '||lastc!=' ')
      putchar(c);
    lastc=c;
  }
}
```

5. 用 for 循环代替例3-3中的 while 循环来计算输入文本中的字符个数。

题解：

```
main( )
{
  int n;
  for(n=0;getcher( )!=EOF;n++);
  printf("%d\n",n);
}
```

6. 编一个程序，找到输入正文中含有元音字母最多的那个单词，输出该单词和它所含元音字母的个数。（一个单词是由一串不含有空格、换行及制表符的字符组成的）。

题解：

```
#define LENGTH 20
main( )
int i,n,maxn;
char c, s[LENGTH],t[LENGTH];
{
  maxn=0;
  c=getchar( );
  while(c!=EOF)
  { while(c==' '||c=='\t'||c=='\n')
      getchar( );
    n=0;
    for(i=0;i<LENGTH && c!=' '&& c!='\t'&& c!='\n'&& c!=EOF;i++)
    { s[i]=c;
      if(c=='a'||c=='e'||c=='i'||c=='o'||c=='u')
          n++;
```

```
        c=getchar( );
      }
      s[i]='\0';
      if(n>maxn)
      { maxn=n;
        for(i=0;(t[i]=s[i])!='/0';i++);
      }
    }
    printf("%d vowel letters in %s\n",maxn,t);
}
```

7. 写一个函数 expand(s1,s2) 将字符串 s1 中形如 a—z 或 0—9 的缩写记号转换成 abc…xyz 或 01…89 的形式，存放在字符串 s2 中，字母允许大写或小写。

题解：

```
expand(s1,s2)     /* expand short-hand notations */
char s1[],s2[];
{
  int err, i, j, last;
  int form, to;
  err=0;                                  /* no error found */
  j=0;                                    /* index of expanded string */
  for(last=0;s1[last]!='\0';last++);       /* index of last position */
  for(i=0; i<=last; i++)
  { if(s1[i]=='-')                        /* it is a -*/
      if(i==0||i==last)                   /* leading or trailing -*/
        from=to='-';
      else if (to !='-')
          {from=to+1;                     /* a-b-c case */
           if(s1[i+1]!='-')               /* avoid two - in a row */
             to=s1[++i];
           else { to=-1;                  /* remember error */
                  ++i;                    /*ignore short-hand */
                }
          } else                          /* last char was a -*/
           {to=-1;                        /* remember error */
            i++;
           }
    else {from=s1[i];                     /* it is not a -*/
          if (i+2>last) to=-1;            /* not enough characters */
```


此为试读，需要完整PDF请访问：www.ertongbook.com