C语言程序设计实验教程

李宏新　陈　斌　编著

中山大学出版社

C语言程序设计实验教程

李宏新　陈　斌　编著

中山大学出版社
·广州·

图书在版编目(CIP)数据

C 语言程序设计实验教程/李宏新,陈斌编. —广州:中山大学出版社,1998.2
ISBN7—306—01096—4

Ⅰ. C ……Ⅱ. 李 …陈… Ⅲ. C 语言—程序设计—实验Ⅳ. TP3

中山大学出版社出版发行
(广州市新港西路 135 号)
英德人民印刷厂印刷　广东省新华书店经销
787 毫米×1092 毫米 16 开本　19.25 印张 456 千字
1998 年 2 月第 1 版　　1998 年 2 月第 1 次印刷
印数:0001—3000 册　定价:26.00 元

前　言

C 语言是一种编译型程序设计语言，它兼顾了多种高级语言的特点，并保留了汇编语言的主要特点。C 语言程序处理功能强，具有良好的移植性，而且可以直接实现对系统硬件及外围接口的控制，具有较强的系统处理能力。

C 语言又是一种结构化程序设计语言，它支持当前程序设计中广泛采用的由顶向下逐步求精的结构化程序设计技术。此外，它具有完善的模块程序结构，从而为软件开发中采用模块化设计方法提供了有力的保障。

在我国，随着计算机应用的深入发展，C 语言的应用正在迅速普及和推广中，它在软件开发工作中占据的地位和所起的作用越来越强。

学习 C 语言程序设计必须上机实习，才能收到应有的学习效果。根据多年的教学体会，我们编写了本书，期望通过它加强在学习 C 语言程序设计中对上机实践的指导，同时也通过实验加强对基本内容的理解、掌握和灵活应用，从而提高学习质量，以及利用 C 语言来解决实际问题的能力。

根据教学要求，本书共编写了 11 个实验。当然，它们只能是最基本的要求。它们包括：在计算机上建立、修改、编译、运行一个 C 语言程序的全过程、输入与输出、各种控制语句、各种数据类型、函数，以及它们在程序中的应用等。每个实验包含实验目的、预备知识、实验内容和实验要求四个部分，遵照循序渐进的规律，为了使学习者更好地理解实验内容，又分例题与学习者自己完成的作业两部分。例题部分已给出简单的分析及完整的程序，读者可以参照教学中涉及的章节，分析、运行、领会例题所给出的程序，以加深对本章节内容的理解。而要求自己完成的题目，正是让读者学以致用、考核自己能力的机会。

全书所给出的预备知识，不是简单的教材内容的重复，而是尽量归纳出各章节的要点，以便供读者在实验中需要时方便查阅。其内容并不局限于哪一本具体教材，而是各章节的基本要点。

本书可作为高等院校本、专科以及中等专科学校 C 语言程序设计课程的实验教材，也可以作为自学者的教材。

本书的第二、五、六、十、十一章由陈斌编写，其余部分由李宏新编写。全书的统稿工作由李宏新完成。本书的修改输入过程中，得到中山大学计算机科学系陈晓蘅、黄红桃、邱梅等同学的大力支持，特此表示感谢。

由于时间仓促，加上编者水平有限，缺点错误在所难免，恳请广大读者批评指正。

编　者

1995 年 12 月

目　录

第一章　运行C语言程序

1.1 实 验 目 的

通过本实验掌握：

1. 语言程序的基本结构；
2. 建立、运行 C 语言程序的基本过程；
3. 书写 C 语言程序的基本要求。

1.2 预 备 知 识

1.2.1 利用计算机处理问题的主要过程

有人以为计算机是“万能”的，人们只要把任务告诉计算机，它就会自动完成一切，并得到正确结果。其实这是一种误解。现在人们所使用的多数计算机系统，是按照人们事先确定的方案，执行人们指定的操作步骤，才能得到所需要的结果。从拿到一个任务到得出正确结果，要经过以下几个主要阶段：

(1) 分析问题。即分析用户需求，用明确的语言，把问题分解，明确描述出要求我们具体要做哪些事情。例如，已经给出哪些数据，需要输出什么样的数据，得到什么形式的结果(包括输出数据本身和数据输出格式)，要进行哪种处理 (如分类排序，或文字检索……)，要用到哪些硬件和软件等。

(2) 确定处理方案。如果是数值计算问题就要建立数学模型。例如要计算人造卫星的飞行轨迹，就必需写出有关的方程，也就是先要将一个物理过程或工作状态用数学形式表达出来。有时数学模型很复杂，可以作一些简化 (在允许误差范围内)，得出近似的计算公式。如果是非数值运算问题，也要确定处理方案。例如对 100 个数排序，由于排序的方法很多，各有优缺点，因此需要先确定采用哪一种排序方法，对同一个问题可以用不同的方案来处理，不同的方案决定了不同的处理步骤，效率也有所不同。

(3) 确定操作步骤。根据处理方案，具体列出计算机如何进行操作的步骤。例如，已经确定用某种排序方法对一批数进行排序，还远远不够，计算机仍然不能立即进行运算，还需要将处理方案具体化，写出解决问题的步骤，这种规定的操作步骤称为“算法”。

(4) 根据选定的算法，用规定的形式画出程序流程图。对照解决问题的步骤，认真核对是否对头。由于流程图的逻辑关系直观，如果解决问题的方法存在问题，能较容易

被查出来。

(5) 根据操作步骤编写源程序。用计算机语言表示出操作步骤就得到计算机程序。一个程序是由若干条计算机指令组成的，它通知计算机一步一步如何执行。换句话说，要使计算机完成所需的处理，必须编写出相应的计算机程序，有时把这一工作称为编码。

(6) 将计算机程序输入计算机并使它运行，如果程序是正确的应能得到预期的结果。如果得不到正确结果应检查前面几步是否有误，改正后再上机运行。

(7) 整理输出结果，写出有关文档资料。

图 1-1 表示计算机处理一个实际问题的主要过程。

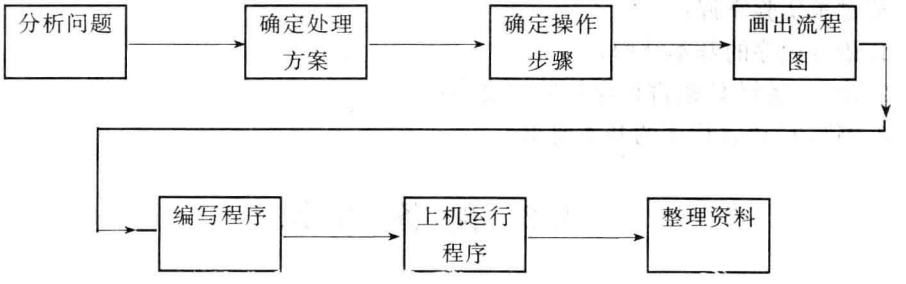

图 1-1 计算机处理一个实际问题的主要过程

以上介绍的是计算机处理一个简单问题的主要步骤，如果处理的是一个庞大复杂的任务，需要采用软件工程的方法。处理的过程要复杂得多，在此不准备详述。

1.2.2 C 语言程序的基本结构

一个完整的 C 语言程序是由一个或多个具有相对独立功能的程序模块结合而成，这样的程序模块通常称之为函数。从格式上来看，每个函数是由函数名和大括号对“{”、“}”包围的若干语句组成的。在组成 C 语言程序的函数中，必须有一个且只能有一个名为 main 的函数，称为主函数。C 语言程序的执行是从主函数开始，主函数中的所有语句执行完毕，则程序执行结束。为了直观地了解 C 语言的结构，下面给出一个程序例子。

[例 1.1] 统计输入文件中，行、单词和字符个数的程序。

```
#include <stdio.h>
main( )
{ int c,nl,nw,nc,inword;
  inword=0;        nl=nw=nc=0;
  while(( c=getchar() )! =EOF )
  {  nc++;
     if ( c=='\n' )  nl++;
     if ( c==' '||c=='\t'||c=='\n' ) inword=0;
       else if ( inword==0 )
        { inword=1;   nw++;  }
```

```
    }
    printf("line=%d word=%d character=%d\n",nl,nw,nc);
}
```

注意：

程序中的 'EOF' 是输入结束符，键盘输入时同时按下 'Ctrl' 键和 'z' 键，然后再打入 '回车' 键。其内部的值是 -1，所以要用 'int' 来说明。

本程序只由一个函数构成，函数名为 main。main 后的圆括号原来是放参数表，此处没有参数，但圆括号不能省去。前面带有 # 的语句是编译预处理语句，以后将专门进行讨论。

C 语言函数的一般格式如下所示：

```
[ 函数类型 ] 函数名( 参数 )
参数说明;
{
  数据说明部分;
  执行语句部分;
}
```

例 1.1 中的

```
int c,nl,nw,nc,inword;
```

是数据说明部分，往下是执行语句部分。

为了更好理解 C 语言程序结构，下面再给出一个程序例子。

[例 1.2] 编写 m 的 n 次方的值的函数，然后调用它来计算 3 的 9 次方的值。

```
main( )
{ int x, y, z;
  scanf("%d%d", &x, &y);
  z=power( x, y );
  printf("%d ^ %d=%d\n", x, y, z );
}
power( x, n )
int x, n;
{ int i, p=1;
  for ( i=1; i<=n; i++ )
    p=p * x;
  return( p );
}
```

本程序由两个函数组成，一个名为 main，一个名为 power。在组成 C 语言程序的函数中，必须有一个且只能有一个名为 main 的函数，称为主函数。除主函数之外的函数由用户命名，如上例中的 power 函数。

C 语言程序的执行是从主函数开始，主函数中的所有语句执行完毕，则程序执行结

束。和其它程序设计语言一样，C 语言程序中也可以使用注释，注释部分格式是：

/* 注释部分 */

注释部分不参加编译，仅供读者阅读方便。

1.2.3 建立、运行 C 语言程序的基本过程

建立，运行一个 C 语言程序的基本过程如图 1-2 所示。

由于建立源程序的编辑程序有很多种，如 UNIX 下的 vi，ed，MS－DOS 下的 edlin 等；对于不同的系统，其编译、连结及运行的命令也各有不同，请读者根据自己的系统参阅有关的手册，认真领会，反复练习，尽快掌握建立、运行 C 语言程的整个操作过程。

1.2.4 书写 C 语言程序的基本要求

C 语言程序的书写式自由度较高、灵活性强、有较大的任意性。但是，为了避免程序书写的层次混乱不清，便于人们阅读与理解，一般用一定格式的书写方法。应该首先声明，这种书写要求并非是计算机要求，而是为了给人们提供便利。归纳起来大致如下：

(1) 一般情况下，每个语句占用一行；

(2) 不同结构层次的语句，从不同的起始位置开始，在同一结构层次中的语句，缩进同样的字数，即同一结构层次中的语句对齐。

(3) 表示层次结构的大括号，写在该结构化语句第一字母的下方，与结构化语句对齐。它可以单独占一行。

(4) 语句中不同单词间可以加以空格，使它们排列得更有规律。

总之，采用这样格式书写的程序，结构层次清晰，充分体现了结构化程序的特点，十分便于阅读和理解，当程序运行有错时，也便于排错。

1.3 实 验 内 容

根据本次实验的具体情况，下面仅给出例题部分。读者可通过运行以下例题，理解、掌握本实验的要求。

[例 1.3] 简单的加法程序

```
main( )                    /*  ---  add of a and b  ---  */
 { int a, b, sum;
   a=123;   b=456;
   sum=a+b;
   printf("sum is %d\n", sum );
 }
```

运行得到结果：

```
sum is 579
```

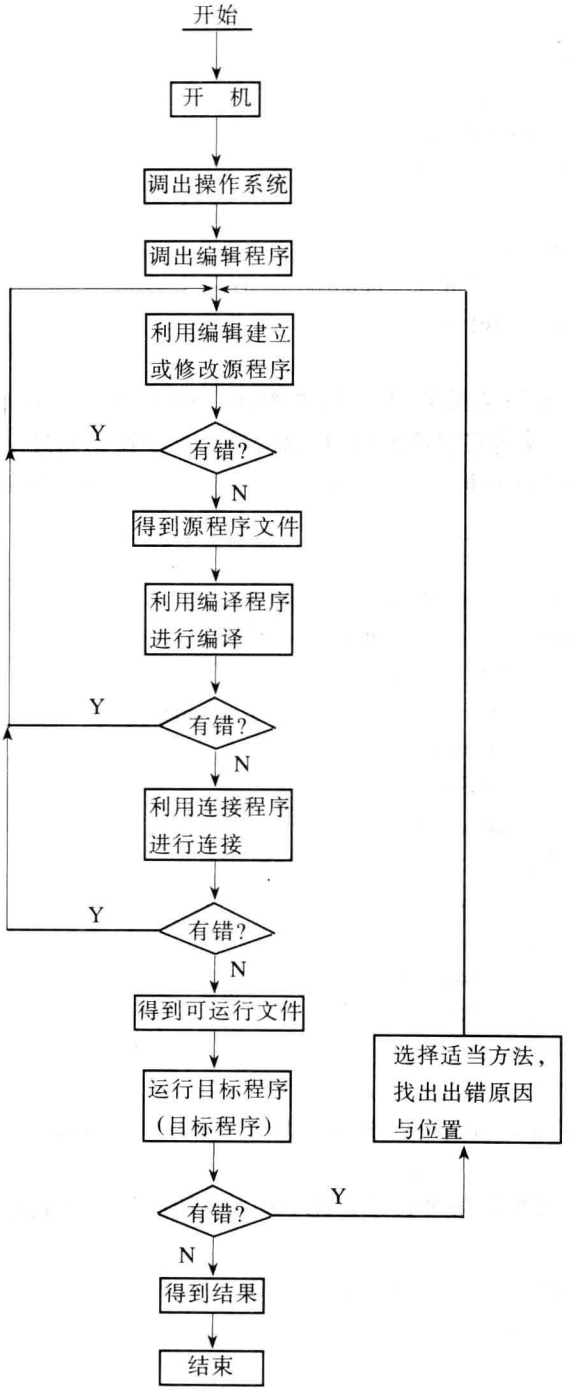
开始
开　机
调出操作系统
调出编辑程序
利用编辑建立
或修改源程序
有错?
Y
N
得到源程序文件
利用编译程序
进行编译
有错?
Y
N
利用连接程序
进行连接
有错?
Y
N
得到可运行文件
运行目标程序
(目标程序)
有错?
Y
N
得到结果
结束
选择适当方法，
找出出错原因
与位置

图 1-2

[例 1.4] 把大写字母转换成小写字母的程序。本程序以字符'/'为结束符号。

```
#include <stdio.h>
main( )
{ char c;
  putchar('>');
  do { c=getchar( );
       if ( ('A'<=c)&&(c<='Z') )
           c=c+32;
       putchar( c );
       if ( c=='\n' )   putchar('>');
     } while (c! ='/');
}
```

[例 1.5] 计算从标准输入(通常指键盘)上输入的字符中每个数字,空白符(空格、制表符和换行)及其它字符出现的次数,设输入字符不超过 1000 个。

```
#include<stdio.h>    /*   count digits, white space, others   */
main( )
{
  int c, i, nwhite, nother;
  int ndigit[10];          nwhite=nother=0;
  for( i=0; i<10; ++i )
   ndigit[i]=0;            /*      array initializers       */
  while (( c=getchar( ))! =EOF )
   if ( c>='0' &&.c<='9' )
       ++ndigit[c-'0'];
     else if(c==' '|| c=='\n' || c=='\t')
             ++nwhite;
           else
             ++nother;
  printf("digits=");
  for( i=0; i<10; ++i )
  printf("%4d", ndigit[i]);
  printf("\n white space=%4d, other=%4d\n", nwhite, nother);
}
```

[例 1.6] 求出比某个数(LIMLT)小的所有素数,并把它们以每行 6 个数的形式输出。

```
/*    printf all primes less than LIMIT    */
#define LIMIT 1000
main( )
{ int j,  k,  cnt=0;
  for ( k=2; k<LIMIT; ++k )
```

```
    { j=2;
      while ( k%j! =0 )    ++j;
      if ( j==k )
        {       /* a prime has found */
          if ( cnt%6==0 )   printf("\n");
          printf("%10d", k);   cnt++ ;
        }
    }
    printf("\n");
    printf("there are %4d prime numbetr less %6d\n", cnt, LIMIT );
}
```

程序运行结束后，应以每行 6 个素数的形式输出所求得的素数。

[例 1.7] 狼、羊、白菜摆渡过河的问题。一个猎人带着一只狼、一只羊和一筐白菜过河，但渡船太小，猎人驾船一次只能带一样东西。由于狼会吃羊，羊会吃白菜，所以狼和羊，羊和白菜不能在猎人不在场的情况下相处。编出程序，求出猎人怎样把它们安全地带到彼岸。

设以字母 P、L、C 和 X 分别表示猎人、狼、羊和白菜，以字母组合表示两岸出现的情况。如在开始时、猎人、狼，羊和白菜都在彼岸，那么，在彼岸一边就有 LCXP 的组合。据题意，可以知道，在整个过程中，两岸始终不允许出现 LC、CX、LP、XP 的组合情况，因此，在此岸所出现的情况只允许是下面组合：LCXP、LXP、LCP、CP。如图 1.3 所示：

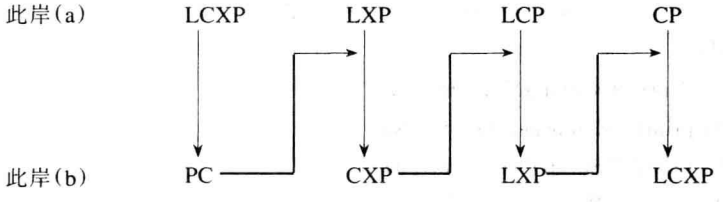

图 1-3

读者可暂不去理会程序实现的算法，待学习了数组后再回过头来研究。下面给出完整的程序。

```
/* ----    file name: lcxp.c   date: 18/12/95 ----    */
main( )
{ int a[4], b[4], p[15];
  int k, i, j, c;        j=0;     p[0]=0;
  for ( i=1; i<=3; i++ )
  { a[i]=i+1;     b[i]=0; }
while ( 1 )
{ /* Repeat run Until a[1]==0 && a[2]==0 && a[3]==0 … */
  for ( i=1; i<=3; i++ )
  { c=a[i]; a[i]=0;
```

```
if ( abs(a[1]-a[2])==1||abs(a[1]-a[3])==1||abs(a[2]-a[3])==1||c==0 )
    a[i]=c;
else break;
  }
a[i]=p[j];   p[++j]=c;
if ( a[1]==0 && a[2]==0 && a[3]==0 )   break ; /* Jmp out */
if ( b[1]==0 && b[2]==0 && b[3]==0 )
{ b[1]=c;  p[++j]=0;  continue;  }
k=0;
for ( i=1; i<=3; i++ )
if ( b[i] ! =0 )
  { k=b[i]; break; }
if ( abs(k-c)==1 )
{ p[++j]=k;  b[i]=c;  continue; }
p[++j]=0;
for ( i=1; i<3; i++ )
if ( b[i]==0 )   b[i]=c;
}

printf("\n\n");
for ( i=1; i<=j; i++ )
{ printf("~~~~~~~~~~~~~~~~~~~~\n\n");
switch ( p[i] )
  { case 0: printf("(Nothing)");  break;
    case 2: printf("(Cabbage)");  break;
    case 3: printf("(Sheep)");    break;
    case 4: printf("(Wolf)");     break;
  }
if ( i%2 )  printf("------------>\n");
  else      printf("<------------\n");
}
printf("~~~~~~~~~~~~\n");
}
/* file name: lcxp.c made date: 18/12/95 */
```

运行程序得到结果输出：

```
~~~~~~~~~~~~~~~~~~~~
(Sheep)------------>
~~~~~~~~~~~~~~~~~~~~
(Nothing)<------------
~~~~~~~~~~~~~~~~~~~~
```