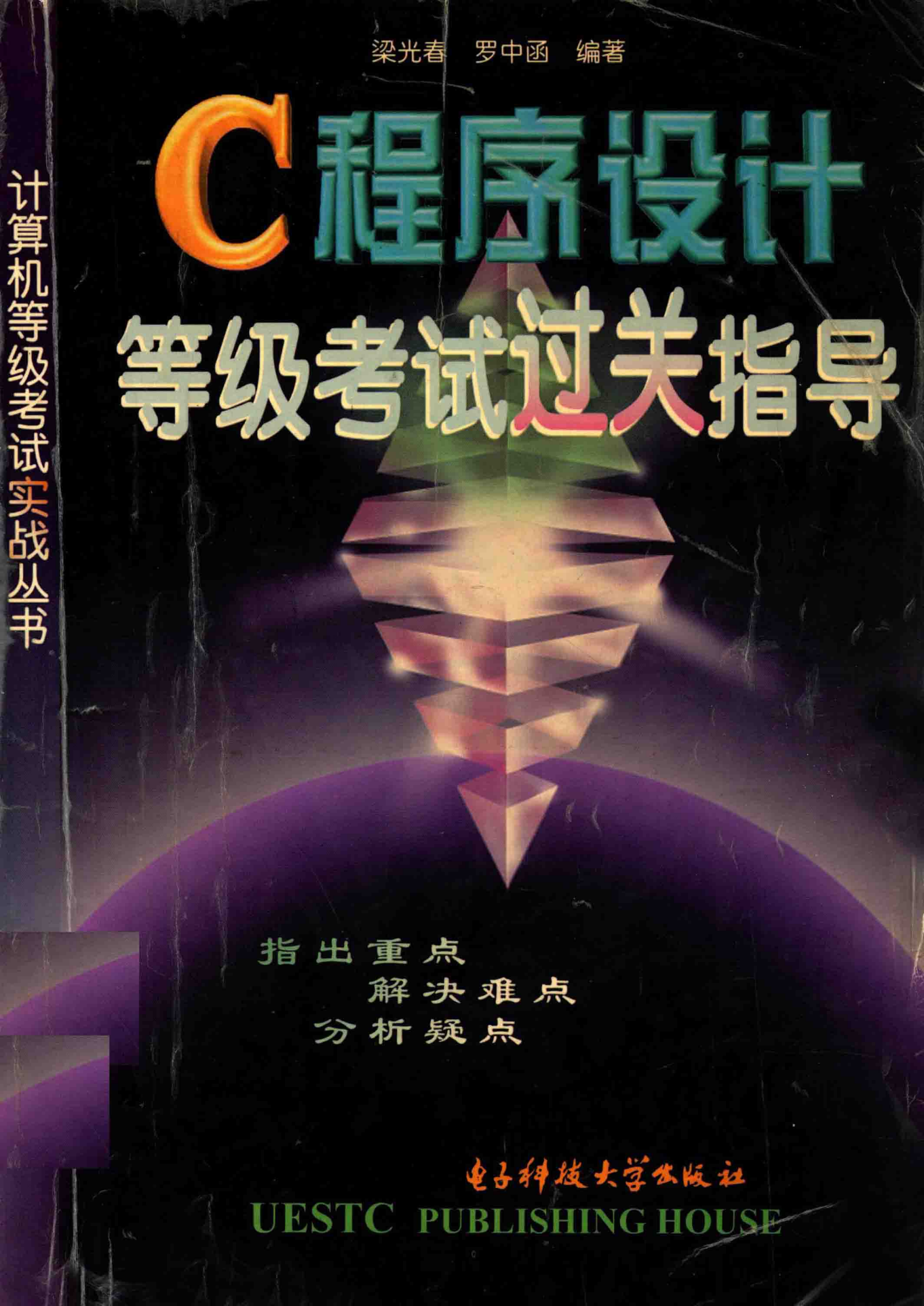
梁光春　罗中函　编著

C程序设计等级考试过关指导

计算机等级考试实战丛书



指出重点
解决难点
分析疑点

电子科技大学出版社
UESTC PUBLISHING HOUSE

●计算机等级考试实战丛书●

C程序设计等级考试过关指导

梁光春　罗中函　编著

电子科技大学出版社

内 容 提 要

本书按全国计算机等级考试大纲的要求，根据作者多次参加四川省C程序设计等级考试命题和阅卷的经验，对考生参加C程序设计二级等级考试（包括一级考试的相关内容）进行有目的的指导，详细地介绍和分析了考试中的重点和难点。书中提供了大量实例以及500道模拟题供考生强化训练。所有试题均按标准题型制作，并附有答案。通过对本书的阅读和练习，将极大地提高考生参加等级考试的应试能力和水平，使考生做到有的放矢，为顺利通过等级考试做好充分准备。

C程序设计等级考试过关指导

梁光春　罗中函　编著

出　　版：电子科技大学出版社　（成都建设北路二段四号，邮编：610054）
责任编辑：张　俊
发　　行：电子科技大学出版社
印　　刷：四川建筑印刷厂印刷
开　　本：787×1092　1/16　印张 14.75　字数 344 千字
版　　次：1998 年 8 月第一版
印　　次：1998 年 8 月第一次印刷
书　　号：ISBN 7—81043—994—4/TP・456
印　　数：1—4000 册
定　　价：18.00 元

前　言

本书主要根据四川省(或重庆市)高等学校非计算机专业计算机等级考试大纲的要求,同时参考了全国计算机等级考试大纲(或其他省市的等级考试大纲)而编写的等级考试参考书。笔者利用历届参加四川省C程序设计等级考试的命题、组题、审题和阅卷的经验,对考生进行有目的的指导。该书系统地简述C程序设计的基础知识和技术。全书共分十一章,每章分别按基本要求、内容提要、典型例题分析、习题与解答等模式来论述C程序设计中的基本概念和技术。在基本要求中提出了掌握、理解、了解等多层次的具体要求;在内容提要中,简单而扼要地提出了本章的基本内容,并突出其重点和难点,通过例子加深基本内容的理解;在典型例题分析中对各种题型中的容易出错或难度较大的问题进行了分析,使考生得到启发;在习题和解答中,提供了各种模拟题,并附有参考解答,供考生练习和强化训练。书中所有的模拟题都是按计算机等级考试标准化命题方式进行的,并经过精心的挑选,内容紧扣大纲,目的明确,全面反映出命题的各种形式和不同的难度,而且具有很高的命中率。

考生通过对这些模拟题的练习,不仅强化和巩固所学的知识而且还熟悉了等级考试中的各种题型,从而提高考生的应试能力和计算机应用水平。

目　录

第十一章 文 件

第一章

概　述

基本要求

1. 了解C语言产生的背景，理解C语言的特点。
2. 掌握C程序的基本结构和C语言程序设计风格。

1.1　内容提要

1. C语言的发展简史及特点

在各种计算机语言中，C语言可以说是较为成功的。C语言是在UNIX操作系统的研制和开发过程中诞生的。在1997年，美国电话与电气公司(AT&T)的贝尔实验室中的丹斯·里奇(Dennis Ritche)先生，在B语言的基础上写成了C语言。1972年投入了使用，他同肯·汤普森(Ken Thompson)于1973年用C语言重写了UNIX操作系统，并且成了以后各种版本UNIX的发展基础。

现在人们广泛地采用C语言来完成各种编程任务，是因它具有下列独特优点：

C语言是一种通用的结构化程序设计语言，用C编写的程序由多个模块(函数)组成，每个模块完成一项功能；C语言的语句简洁、灵活，使用起来得心应手，它一共只有32个关键字，很便于初学者掌握；C语言具有丰富的运算符，使用起来简单精炼，生成的机器代码质量高，内存占用少，运行速度快；C语言的数据类型丰富，可进行复杂的数据结构运算，它的指针数据类型的使用更有特点，它使参数传递简单、迅速、节省内存；C语言具有位(bit)操作能力，可直接对计算机硬件的物理地址进行访问，它既是高级语言又兼有低级语言的功能；同时，C语言具有较好的可移植性。

2. C程序的基本结构

构成C程序的几个基本部分：

```
main( )
  {
   变量定义
   程序语句
  }
```

变量定义部分定义所有需要的变量，程序语句部分包括C的各种基本语句、控制语句及函数调用，每个程序语句和变量定义以分号(;)结束。由于C语言中的所有程序模块都是按函数来处理的，故任一函数的结构也应是这种形式。

程序开始处的 main 表示主函数，它后边的小括号表示函数。一个 C 源程序可有多个函数但只能有一个主函数，且必须有一个主函数，在 main( )下面的一对大括号内是函数体，所以函数体是以大括号开始到大括号结束。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。

3. 设计 C 程序的基本步骤

C 程序设计的一般步骤，如图 1-1 所示。

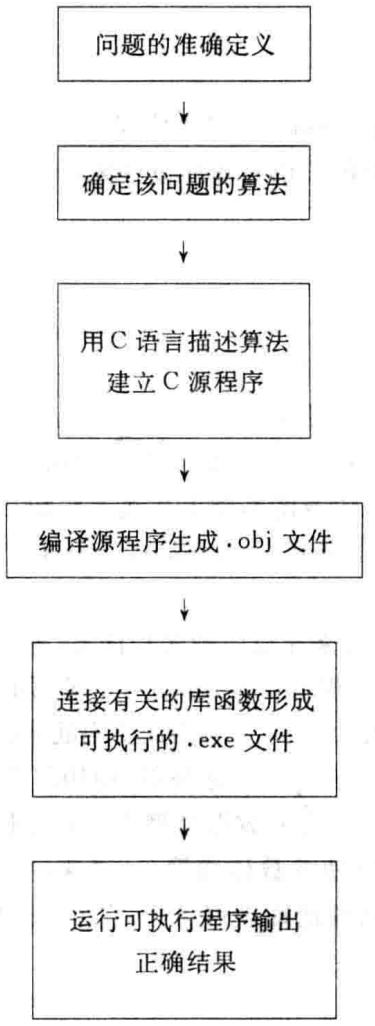


图 1-1

图 1-1 中，步骤一和步骤二是用计算机解决问题的基础。如果在解决的问题不明确，或者在解决问题的方法不确定时就开始编程序必然会造成事倍功半的结局。一旦完成上述两个步骤后就可用 C 语言的语句来描述解决该问题的算法；然后建立 C 源程序，在计算机内形成 C 文件。计算机不能直接执行 C 源程序，它必须通过 C 语言的编译器把它转换成机器语言的程序，形成 .obj 文件。在 C 程序中使用了各种类型的库函数(或者标准函数)，要形成可执行程序，就需将这些库函数(或者标准函数)与目标程序(.obj 文件)连接。然后执行程序输出结果。

当一个 C 程序设计经过以上几个步骤(可能反复多次)的工作，最后得到了符合设计要

求的正确结果时，则C程序的设计宣告完成。

4. C语言程序设计的风格

良好的编程风格有助于对程序的阅读、理解和记忆。在学习C程序设计的过程中应养成正确和良好的习惯，在用C语言编写程序时，应该遵循以下的一些原则：

(1)C语言的变量名、函数名和关键字一律都采用小写字母，而符号名和符号常量用大写字母来定义。

(2)程序中的各语句组采用缩进编排，使程序模块和程序块变得十分明了。

(3)在定义变量名时应与它的功能联系起来，可以提高程序的可读性。

(4)充分利用注释语句，将程序的有关信息以注释方式加到程序中，便于记忆和阅读。

(5)在程序中适当地加上空格和空行使程序更加清晰。

(6)用一对花括号“{}”标志程序功能的开始和结束，一般情况下总是将“{”与“}”按列对齐。

1.2 典型例题分析

1. 单项选择题

C语言具有低级语言的能力，主要指的是：(　　)

A. 程序的可移植性。

B. 具有控制流语句。

C. 能直接访问物理地址，可进行位操作。

D 具有现代化语言的各种数据结构。

分析：

选择项A、B、D列出的特性都是高级语言的功能，唯有C才是低级语言的特性，故应选择C项。

答案：C

2. 多项选择题

下列叙述中有哪些是编程风格正确的描述(　　)。

A. 大小写字母用在不同的场合，一般除了符号名和常量名用大写字母外，其他一律用小写字母。

B. 程序中的语句一律都采用缩进的编排格式。

C. 程序中的注释可有可无。

D. 使用有意义的标志符。

E. 使用括号来改善表达式的清晰度。

分析：在所有选择项中除了C外都是编程风格的正确描述，提倡在C程序中多增加注释，增加其程序的可读性。并非注释是可有可无。故正确的答案有：A、B、D、E。

1.3 习题与题解

1.3.1 习题

1. 单项选择题

(1)不正确C程序的描述(　　)。

A. 每个语句和数据定义的最后必须有个分号。

B. 一个C程序总是从main( )函数开始执行。

C. C程序的书写格式要求严格,一行内只能写一个语句。

D. C语言的本身没有输入输出语句。

(2)选出C语言正确结构形式的描述(　　)。

A. main( )函数必须放在程序之首。

B. 每一行内只能写一个语句,每个语句之后必须有分号。

C. C语言没有输入输出语句。

D. C程序的自由度大,可以从任何函数执行。

(3)合法的变量名(　　)。

A. 3X

B. auto

C. _123

D. &y

2. 多项选择题

(1)指出下列程序的结构哪些是正确的(　　)。

A.
```
fun( )
{
  …
ioa( )
{
  …
 gon
  {
    …
  }
}
```

B.
```
fun( )
{
  …
}
gon( )
{
  …
}
ioa( )
{
  …
}
```

C.
```
fun( )
{
  …
gon( )
{
  …
}
ioa( )
{
  …
}
```

D.
```
main( )
{
  …
}
fun( )
{
  …
}
gon( )
{
  …
}
```

E.
```
fun( )
{
  …
}
main( )
{
  …
}
gon( )
{
  …
}
```

(2)以下哪些是C语言具有的特点(　　)。

A. 具有现代化语言的各种数据结构。

B. 生成的目标代码质量高，程序执行效率高。

C. C 语言没有灵活性，所以容易掌握，应用面广。

D. C 语言的可移植性好。

E. 运算符丰富。

(3)所有的函数都是平行的，其主要的理由是：(　　)。

A. 定义函数时，互相是独立的。

B. 一个函数都不能从属于另一个函数。

C. 函数可以嵌套定义，但不能互相独立。

D. 任何函数都不能调用 main( ) 函数。

E. 函数不可以嵌套定义，但可以嵌套调用。

1.3.2 习题解答

1. 单项选择题

(1)C　　(2)C　　(3)C

2. 多项选择题

(1)B,D,E　　(2)A,B,D,E　　(3)A,B,D,E

第二章

基本数据类型

基本要求

1. 了解C语言中有哪些数据类型。
2. 理解C语言中基本类型的常量、变量以及字符串常量的概念。
3. 掌握C语言中的常量、变量以及变量存储类别的灵活应用。

2.1 内容提要

1. C语言的基本数据类型及数值范围

(1)整型数据

整型常量：凡是不带小数点或指数的数是整型常量，其范围－32768～＋32767(即两个字节)。

整型变量：存放整型数据的变量是整型变量，可以分为短整型、长整型和无符号整型，其数值范围－32768～＋32767。

整型变量说明的一般形式：

int　变量名表

(2)字符型数据

字符型常量：用单引号引起来的字符叫字符常量，用一个字节来表示。

字符型变量：存放字符型数据的变量是字符型变量。

字符型变量说明的一般形式：

char　变量名表

(3)实型数据

实型常量：凡是带小数点或指数的数是实型常量，具有7位有效数字的实型常量叫单精度实型常量(占4个字节)，具有15～16位有效位数字的实型常量叫双精度实型常量(占8个字节)它们的数值范围分别为 10^{-38}～10^{38}和 10^{-308}～10^{308}。

实型变量：存放实型数据的变量是实型变量，相应地分为单精度实型变量和双精度实型变量。

实型变量说明可分为两种形式：

a. 单精度实型变量说明的一般形式：

float 变量名表；

b. 双精度实型变量说明的一般形式：

double 变量名表；

2. 常 量

C 语言中除了上面讲的基本类型常量以外还有其他的一些常量，例如二进制常量、八进制常量、十六进制常量、字符串常量、反斜线字符串常量。

(1)八进制常量

用八进制表示的常量，并以 0 开头的数表示八进制数，例，0123(即$(123)_8$)表示 123 是一个八进制的数。

(2)十六进制常量

用十六进制表示的常量是十六进制常量，并以 0X 开头来表示一个十六进制的常量。例，0X123(即$(123)_{16}$)是一个 16 进制的数据。

(3)字符串常量

用双引号括起来的字符序列叫字符串常量，例如"a"，"abc"都叫字符串常量，注意：用双引号括起来的串，它包含了一个隐含字符 '\0'(NULL)。

(4)反斜线字符常量

C 语言规定了一系列反斜线字符串常量，每个反斜线字符串常量都代表了特殊意义，在程序中用反斜线字符串常量替代了等价的 ASCII 码，增加了程序的可移植性，反斜线字符串常量又称转义序列。表 2-1 是转义序列表。

表 2-1

反斜线码	意义(ASCII 码)	十六进制	键
\0	空(NULL)	0X00	Ctrl/L
\a	报警(BEEF)	0X07	Ctrl/G
\b	退一格(BS)	0X08	Ctrl/H
\t	水平制表(HT)	0X09	Ctrl/I
\n	换行(LF)	0X0A	Ctrl/J
\v	垂直制表(VT)	0X0B	Ctrl/K
\f	走纸换页(FF)	0X0C	Ctrl/L
\r	回车(CR)	0X0D	Ctrl/M
\"	双引号(")	0X22	"
\'	单引号(')	0X27	'
\?	问号(?)	0X3F	?
\\	反斜线(\)	0X5C	\
\ddd	三位八进制数(ddd)		
\xd	十六进制数(d)		

3. 变 量

值可以改变的量称为变量，由于 C 有不同的数据类型，相应就有不同类型的变量，并且

C 语言规定在 C 程序中变量先定义，然后才使用，否则在编译时将会出错。

在不同的位置上定义的变量是具有不同性质的，C 语言中有三个不同的位置定义变量，一是在函数体内定义的变量，这种变量叫局部变量，二是在函数名后小括号内定义的变量，这种变量也是局部变量。

(1)局部变量

在函数体内部，就是函数大括号内说明的变量为局部变量。该变量仅在函数体内有效，超出该函数就自动消失，也就无效了。若再使用它就会出现语法错误(因该变量无定义)。

【例 2.1】
```
fu (void)
 {
 int x ,y;
 char ch;
 }
```

整型变量 x，y 和字符型变量 ch 都是局部变量，它们的作用范围仅在函数 fu()内有效

(2)全局变量

在所有函数外部定义的变量为全局变量，它对所有的函数都有效，作用范围是所有的函数。

【例 2.2】
```
int x, y;
char ch;
fu1( )
{
  int m,n;
  float z;
   ⋮
}
fu2 ( )
{
  int l, x;
  char c;
}
```

x、y、ch 是全局性变量，在 fu1( )和 fu2( )函数内都可引用它。而 m、n、z、l、x、c 是局部变量，m、n、z 仅在 fu1( )内引用它，而不能在 fu2( )内引用它。同样 l、x、c 仅在 fu2( )内引用，不能在 fu1( )内引用。因为超出了它们的作用范围。

(3)局部变量和全局变量同名的冲突问题

在定义局部变量和全局变量时，变量名可以同名，同一个名字在不同作用域内取相应的值是不会混淆的。不用编程者担心，编译程序会自动处理。但是有一点必须注意，全局变量和局部变量发生冲突时，编译器确定局部变量有效(即取局部变量的值)而全局变量无效(不能取全局变量的值)。

【例 2.3】
```
main( )
{
```

```
int x=1;
  {
    int x=2;
    {
      int x=3;
      printf("%d\n",x);
    }
    printf("%d\n",x);
  }
  printf("%d\n",x));
}
```

该程序的运行结果：

```
3
2
1
```

4. 变量的存储类别

编译器在编译C的源程序时，根据存储类别说明符来存放变量，C语言提供了外部变量说明符、静态变量说明符、寄存器变量说明符和自动变量说明符等四种说明符。

(1)外部变量说明符 extern

一个大型的源程序有很多源文件，在一个源文件中定义了全局变量，其他源文件中要引用这个全局变量应在变量说明符前加上 extern 存储类别说明符。

【例 2.4】

f1.c

```
int x,y;
char ch;
main( )
{
⋮
}
f3( )
{
⋮
x=15;
}
```

f2.c

```
extern int x,y;
extern char ch;
 f4( )
 {
 x=y%10;
  ⋮
 }
  f5( )
  {
  ⋮
  y=20;
  }
```

在 f1.c 源文件中说明了两个全局整型变量 x，y 和一个字符型变量 ch，而 f2.c 源文件要引用 f1.c 中的三个全局变量，那么，在 f2.c 源文件的说明中，冠以 extern 关键词，它将告诉C编译器不再为这些全局变量分配存储空间，在连接时把这些变量直接装配就行了，使外部变量的作用域在整个源程序中有效。