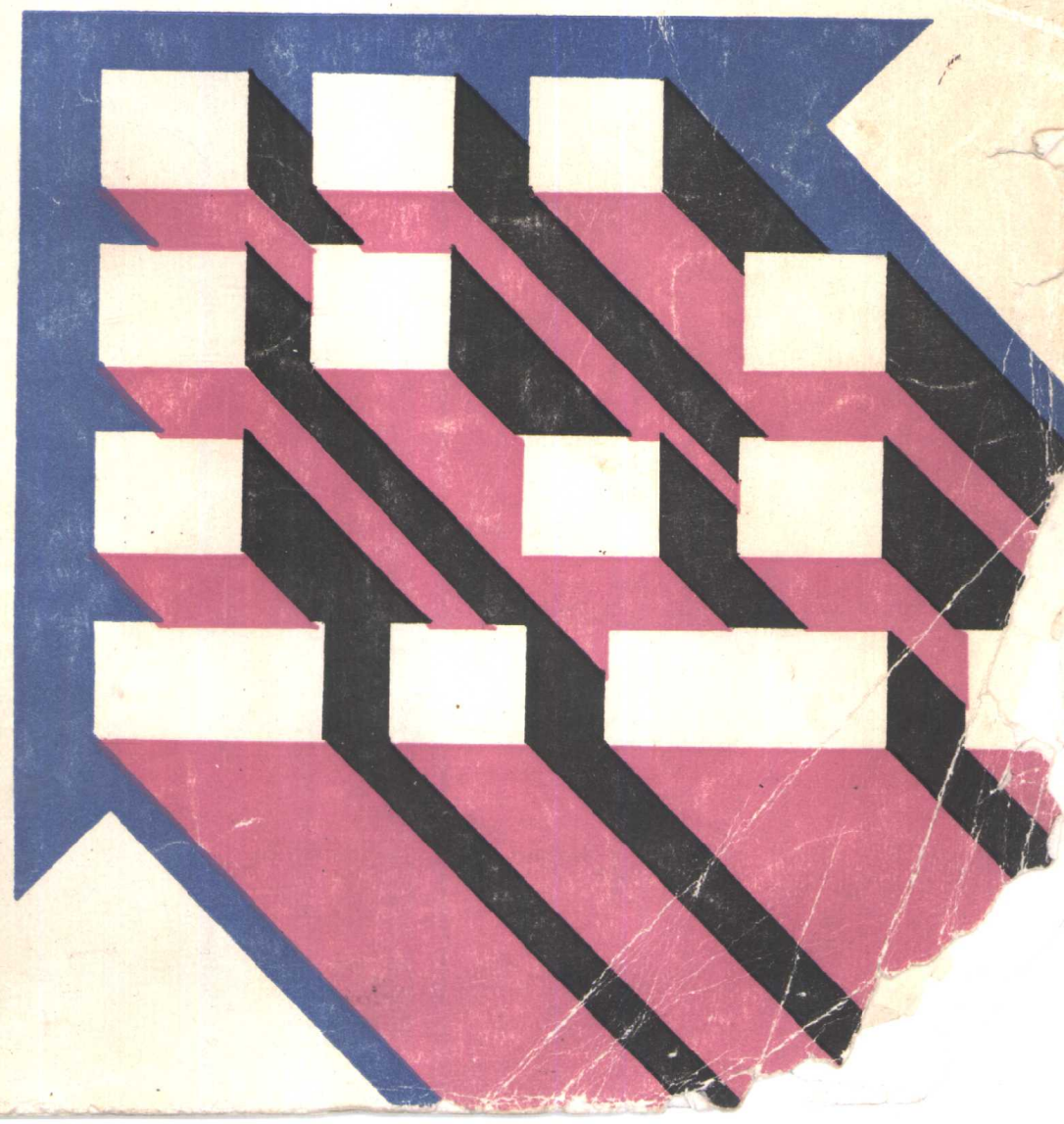
高校计算机等级考试辅导丛书

（非计算机专业）

C语言（二级）

郑志毅　龚沛曾　编著

上海科学技术出版社



高校计算机等级考试辅导丛书

（非计算机专业）

C 语 言（二 级）

丛书主编　杭必政

丛书副主编　王春森　吴永明　乔沛荣

编　　著　郑志毅　龚沛曾

审　　阅　王春森　杭必政

上海科学技术出版社

高校计算机等级考试辅导丛书
（非计算机专业）
C语言（二级）
丛书主编　杭必政
丛书副主编　王春森　吴永明　乔沛荣
编　　著　郑志毅　龚沛曾
审　　阅　王春森　杭必政
上海科学技术出版社出版、发行
（上海瑞金二路450号）
新华书店上海发行所经销　常熟市印刷六厂印刷
开本 787×1092　1/16　印张10.25字数 236,000
1995年9月第1版　1995年9月第1次印刷
印数　1-6,000
ISBN 7-5323-3970-X/TP·55
定价：11.60元

内 容 提 要

本书是高等院校计算机等级考试(非计算机专业)辅导丛书之一。旨在帮助大学生通过C语言等级考试。全书以应试为特色,主要内容包括C语言知识要点、常用算法、习题与答案等。其中知识要点着重于对C语言基本知识的理解,常用算法着重于解题分析,习题与答案着重于检验对基本内容和方法的掌握。

本书编写紧凑、重点突出。可作为高等院校计算机等级考试的辅导用书,也可作为广大计算机用户学习C语言的参考书。

序

计算机的发展推动了社会经济、文化和军事等方面的飞跃发展，而经济等的发展又对计算机的发展提出了更高的要求。我国许多地方早就把计算机列为重点发展的高新技术支柱产业。高校中95%以上的非计算机专业大学生是我国未来计算机应用的最重要的生力军。非计算机专业大学生计算机基础教育的状况，直接影响着我国今后的计算机应用水准。计算机应用知识和应用能力已成为现代大学生知识结构中的重要组成部分。目前各高校学生争选计算机课程，渴求增加计算机知识和增强计算机应用能力，他们已领悟到在信息时代里，没有计算机应用知识等于是新时代的“文盲”。

四年前上海市高教局率先决定建立上海高校非计算机专业大学生计算机应用知识和应用能力等级考试制度，顺应了计算机发展的浪潮。在这以后，北京、陕西、四川、江苏和浙江等全国多数省市已先后开展或准备开展非计算机专业大学生的这种等级考试。从上海和全国有关省市已经进行的这种等级考试来看，等级考试在目前确实促进了课程体制的改革，实验设备的添置和更新，促进了教材和教学手段的改革。

为了帮助非计算机专业大学生进一步提高计算机编程、设计等应用能力，也为了具体指导大学生如何更好地应试等级考试，我们从上海高校中请来了长期在第一线从事计算机基础教学，且有较丰富教学经验的教师编写了这套丛书。

编者以上海等地区的“计算机等级考试大纲”为依据，着重从知识要点、试题分析、习题和解答等方面来组织编写这套《高校计算机等级考试辅导丛书》。我们相信，该套丛书对广大学生会有所裨益。丛书包括：《计算机基础》(一级)、《FORTRAN 语言》(二级)、《C 语言》(二级)、《PASCAL 语言》(二级)、《TRUE-BASIC 程序设计》(二级)、《硬件》(三级)和《软件》(三级)等七本书。该丛书由杭必政任主编，王春森、吴永明、乔沛荣任副主编。史济民、高传善、章鲁、杭必政和王春森分别审阅了有关书稿。

上海科学技术出版社大力支持本套丛书的出版，上海市有关高校领导在本书的编写过程中也给予了积极关心和支持，在此一并表示感谢。由于时间紧迫，书中定有不足和错误之处，敬请读者批评和指正！

主　编

1995年4月

前　言

C 语言是目前在国内外得到广泛使用的一种程序设计语言，既适合编写应用软件又适合编写系统软件。

C 语言概念丰富、功能很强、使用灵活，初学者要完整地掌握它并参加高校计算机 C 语言等级考试是有一定困难的。

本书以《上海高校非计算机专业学生计算机应用知识和应用能力等级考试》考试大纲为依据，等级考试 C 语言试题为蓝本，结合初学者学习中的薄弱环节及应试中的错误之处加以编写。

全书主要内容是知识要点、常用算法、习题与答案三大部分。知识要点从数据、运算和 I/O 操作及控制结构三方面加以阐述，阐述方法是循序渐进与前后交叉相结合、知识归纳与疑点、难点、重点分析相结合，以帮助读者加深对有关知识的理解。常用算法涉及八个方面，包括若干初等数学问题、方程求解、矩阵、排序、查找、字符串、表处理和文件，按等级考试要求格式组织例题，着重分析解题的思路和方法，以帮助读者灵活使用 C 语言，习题部分中的题目是精心选择和组织的，以帮助读者进一步理解、巩固所学的 C 语言基本内容和解题方法。

本书内容由郑志毅、龚沛曾编写，王春森、杭必政审阅。作者期望本书对提高读者 C 语言程序设计能力和应试能力将有所裨益。由于作者水平有限，书中不足之处在所难免，敬请广大读者批评指正。

作　者

1995 年 3 月

目　录

第一章　数　据

第二章　运算和I/O操作

第三章　控 制 结 构

第四章　常 用 算 法

第五章 习题与参考答案

第一章 数 据

一个完整的C语言源程序，从文件角度讲，可以由一个或多个源文件组成，而每个源文件，一般至少包含一个函数，即由一系列变量或函数组成。从函数角度讲，C源程序有且仅有一个名为main()的主函数和若干个函数(可以是零个)组成，这些函数是并列的，不能是嵌套的。

不管主函数在什么位置，程序总是从其开始执行，其它函数通过主函数或别的函数调用而执行。被调用的函数如定义在调用函数之后或不在同一源文件，则要对被调函数加以说明。

C语言知识要点分数据、运算和I/O操作及控制结构三章加以阐述。阐述方法是循序渐进与前后交叉相结合，知识介绍与疑点、难点、重点分析相结合，以有别于教科书。

数据是计算机处理的基本对象，分为常量和变量两种，常量、变量有类型及表示问题。常量的类型仅涉及数据类型，而变量的类型既涉及数据类型又涉及存储类型。存储类型引出变量的生存期和作用域(有效期)。

第一节 基 本 类 型

基本数据类型有字符型、整型和实型(浮点型或单精度型及双精度型)等。

一、常量

(一)字符型

字符型常量又分字符常量和字符串常量。

1. 字符常量

字符常量是指在一对单引号中的一个字符(可打印字符)或反斜杠开始的一个转义表示。如'a'表示字符a;'\n'表示换行符,'\\'表示反斜杠符\,'\''表示单引号';'\101'表示ASCII值为八进制101的字符A,特别'\0'表示其ASCII值为八进制0的null字符;'\x41'表示ASCII值为十六进制41的字符A。

2. 字符串常量

字符串常量是用双引号括起来的一串字符，可以是零个字符。如"china",""。若要表示字符串I'm a student.，它的字符串常量为"I\'m a student."这儿双引号只是字符串的定界符，串中的字符个数为字符串的长度，\'表示字符单引号'。在存储时，系统自动在字符串的末尾加一个字符串结束字符'\0'，俗称字符串结束标志。在内存中，字符串常量又表示其存储首地址。因此"china"既表示字符串常量又表示其存储的首地址，即指向第1个字符c的地址。

读者注意:'a'与"a"是完全不同的。前者是一个字符常量用整数(字符a的ASCII值)存储，后者是一个字符串常量，为存储地址，所指的第1个字符为a，接着1个结束标志'\0'。

(二)整型

整型常量分长整数、无符号整数和整数,分别在整型常数后加 l 或 L、U 或 u 或不加后缀。从数制来讲又有八进制、十六进制和十进制三种,分别是在整型常数前加数 0、数 0x 或不加前缀。如

32767	十进制整数
32767l	十进制长整数
0101	八进制整数
0x41	十六进制整数
65u	十进制无符号整数

外部形式 0101、0x41、65u 在 PC 系列机内均占 2 个字节且值相同。外部形式 32767 与 32767l 虽然其值均为 32767,但在机内所占的字节是不同的,前者为 2 个字节,后者为 4 个字节。

(三)实型

实型常量在表示上又分一般形式和指数形式。

一般形式由整数部分、小数部分、小数点及+、-符号组成。其中整数部分或小数部分可以不出现。正数符号可以省略。如-12.5,.123,456.,+0.5。

指数形式为

一般形式 e 符号整数

或 符号整数 e 符号整数

如-12.5e+3,456.e-3,94e2,1.0e-5

在 PC 系列机中均占 8 个字节。

(四)符号常量

符号常量是一个标识符,它通过宏定义

#define 标识符 常数

定义的,用以代替所确定的常数。如在系统的头文件 float.h,math.h 中就有

#define M_PI 3.14159265358979323846

#define FLT_MAX_10_EXP +38

#define SW_OVERFLOW 0x0008

#define FLT_MIN 1.17549345E-38

二、变量

变量的定义形式为

[存储类型] 数据类型 标识符;

存储类型是变量存储方式的描述,决定生存期和作用域。C 语言中有四种:auto、register、static 和 extern。

auto 存储类型的变量存放在动态存储区。在函数或分程序内说明时,auto 通常可以省略。生存期与作用域仅为函数内或分程序内。

static 存储类型的变量存放在静态存储区。在函数外部说明的称为静态外部变量,可以被函数公用,而在函数内部说明的称为静态内部变量,只能在函数内使用,但不管哪种静态

变量其生存期为整个程序的一次执行。

extern 存储类型的变量也存放在静态存储区，它只在函数外说明，称为外部变量，生存期也是整个程序的一次执行。

静态外部变量与外部变量的区别在于静态外部变量只局限于在说明它的某一个源文件中可见(作用域)，其它源文件不能用任何方式访问它，而外部变量则可以被整个程序所访问。

为了说清楚外部变量的作用域，把前述叫做定义外部变量，而在别处对外部变量的阐述交待叫做说明外部变量。

extern 在定义外部变量时不用，而在说明时不能省。一种是在引用先于定义时，在引用前要说明；另一种是在某源文件中定义的外部变量，其它源文件中要引用，则引用前也要加以说明。如

```
main()
{extern int i;
 printf("%d",i);
}
int i=2;
```

因此，外部变量进行必要说明后，它的作用域可为整个程序。

register 存储类型的变量存放在 CPU 的通用寄存器中，它是为了提高程序的执行速度而引入的。一般只有少数几个变量(也有编译不分配的)可说明为寄存器变量，超过时，作 auto 型变量处理。

变量的数据类型是为了限定变量的取值范围和能施于的操作。不同机型上的基本数据类型所占字节数不尽相同，PC 系列机上基本数据类型如表 1-1 所示。

表 1-1　基本数据类型

类型	所占字节	数值范围
char	1	−128～+127
unsigned char	1	0～255
short	2	−32768～32767
unsigned short	2	0～65535
int	2	−32768～32767
unsigned int	2	0～65535
long	4	−2147483648～2147483647
unsigned long	4	0～4294967295
float	4	−1.0e38～1.0e38 7 位精度
double	8	−1.0e308～1.0e308 16 位精度

由于(基本)整型 int 的数在机内以补码形式表示，故 32767+1 机内表示的外部形式为−32768。读者在使用 short、int、long 型时要特别注意数值范围。

变量定义中的标识符是变量名，是以字母或下划线开头，由字母、数字和下划线组成。

在 ANSI C 中，变量名允许长度为 31 个字符，如

```
int member_of_class;
float _f;
double d;
```

变量的地址是指其所占字节的首地址，用

```
&变量名
```

表示。如 &d 表示 double 型变量 d 的首地址。

注意，寄存器变量是不允许取地址的。

第二节 指　　针

指针是C语言中一个重要的数据类型，它是有关一个对象的地址的变量，对象可以是各种类型的变量、数组、数组元素、函数、一片连续存储单元，甚至还可以是另一指针。使用指针可以实现间接访问、动态存储分配和对函数实参所指对象内容的修改。人们改变指针所含的地址，就可以处理相应位置里的信息。

指针本身所需的存储空间对 PC 系列机是 2 个字节，有的编译系统是 4 个字节，而其所指对象可以是若干个字节。

一、指针的定义

一般形式为

```
存储类型　数据类型　*标识符;
```

这是一级指针定义，标识符为指针名，存储类型是指针本身的存储类型，通常是 auto、static 和 extern 之一，它们决定指针的生存期和作用域，含义类似于变量的存储类型。数据类型是指针所指对象的数据类型。如在函数内定义

```
auto int *p;
```

则 p 为自动类型指针变量，指向整型数据，可以是变量、数组元素。又如在函数外定义

```
double *g;
```

则 g 为外部类型指针变量(extern 不用)，指向 double 型数据。

当数据类型为结构、FILE 时，分别称为指向结构的指针和文件指针(详见下面几节)。

当数据类型为 void 型时，则指针不指向具体的数据类型，这类指针用于与指向任一目标类型的指针的转换。在动态存储分配里经常用到指针类型转换。

与一般变量一样，指针定义仅仅分配指针本身的存储，其内容即指向目标的地址并没确定。

二、多级指针的定义

一般形式为

```
存储类型　数据类型　**标识符;
```

这是二级指针定义。存储类型仍为二级指针本身的存储类型，它指向一级指针，而数据类型是一级指针所指对象的数据类型。标识符为指针名。二级指针占有的两个字节用于存放其所指的一级指针的地址。如

```
double  ** p;
```

则 p 为二级指针，它指向一个指针，该指针指向 double 型数据。俗称 p 为指向 double 型数据的二级指针。

当又有

```
double  * g,d;
    p=&g;g=&d;
```

时，二级指针与数据间的联系示意图如图 1-1 所示：

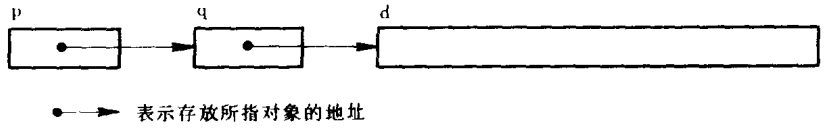


图 1-1 二级指针与数据联系

类似，指向二级指针的指针称为三级指针，定义时只要在标识符前加三个 * 。

三、其它指针的定义

指针除可以指向一个变量外，还可以指向含有 n 个元素的一维数组，俗称数组指针；指向一个函数，俗称函数指针；指针也可以组成数组，称为指针数组。

(一)数组指针定义

一般形式

```
存储类型  数据类型  (* 标识符)[n];
```

存储类型的含义同前，标识符为指针名，它指向含有 n 个数据类型元素的一维数组，n 可为整型常量表达式。如

```
int  (* p)[n];
```

则 p 为指向含有 n 个整型数据元素的一维数组的指针，n 是符号常量或整型常量表达式。

(二)函数指针定义

一般形式

```
存储类型  数据类型  (* 标识符)();
```

存储类型含义同前，数据类型为指针所指函数的类型，标识符为指针名。如

```
char  (* f)();
```

则 f 为指向数据类型为 char 型的函数指针。

(三)指针数组定义

一般形式

```
存储类型  数据类型  * 标识符[整型常量表达式];
```

其中标识符为指针数组名，存储类型为指针数组的存储类型，元素个数由整型常量表达式确定，每一个数组元素为指针，指向所述数据类型的数据。如

```
int  * p[5];
```

则 p 为指针数组，含有 5 个元素，每个元素为指向整型数据的指针。

第三节　构 造 类 型

构造类型是由基本类型、指针类型或构造类型组成。主要有数组、结构、联合和枚举。数组是一种同一类型数据的组织；结构是一种多种类型数据的组织，而联合是一种可以在不同的时间内拥有不同类型和不同长度的组织；枚举是只能取若干值的一种数据类型。

一、数组

数组定义的一般形式为

存储类型　数据类型　数组名[整型常量表达式 1]…[整型常量表达式 n]；

当[整型常量表达式]只有一个时，称为一维数组，两个称为二维数组，以此类推。其值为该维元素个(行、层)数。

存储类型通常为 auto、static 和 extern 之一。如

```
static int a[5];
```

定义 a 是一个静态整型数组，有 5 个元素 a[0]、a[1]、a[2]、a[3]和 a[4]。若在函数内定义，则是静态内部数组，否则是静态外部数组。

对于不作形参的数组定义，系统将为其分配连续存储区，字节数为元素类型所占字节数乘以元素个数，数组名为其所占存储区的首地址，是一个确定的常值。因此对上述静态整型数组 a，a＋＋是错误的。对作形参的数组，则数组名是一个可变地址。如命令行参数里的形参：字符指针数组 argv[]是允许 argv＋＋的。

二、结构

结构类型说明的一般形式为

```
struct  结构名
{数据类型        成员名 1;
 数据类型        成员名 2;
    …
 数据类型        成员名 n;
};
```

此后

```
struct      结构名
```

就是一种结构类型，可类似于基本类型中的 char、int、float 和 double 加以使用，需占字节数可由 sizeof(struct 结构名)算出。

说明了结构类型，系统并不为其分配存储空间，只有定义了此结构类型的变量、数组和指针等，才为变量、数组和指针分配存储单元。如

```
struct student{
  long num;
  char name[20];
  float score;
```

```
}st,stclass[30], * p;
```

或

```
struct student{
  long num;
  char name[20];
  float score;
 };
  struct student st,stclass[30], * p;
```

或

```
struct{
  long num;
  char name[20];
  float score;
 }st,stclass[30], * p;
```

是等价的定义形式。

结构类型 struct student 占28个字节，系统为结构变量 st 分配28个字节，为结构数组 stclass 分配28×30个字节，为指向结构 struct student 的指针 p 分配2个字节。

结构成员的数据类型也可以是结构，形成结构的嵌套定义。如

```
struct student{
  logn num;
  char name[20];
  float score;
  struct address{
        int post;
        char * addr;
        char tel[10];
       }add;
}st,stclass, * p;
```

或

```
struct address{
  int post;
  char * addr;
  char tel[10];
 };
 struct student{
  long num;
  char name[20];
  float score;
  struct address add;
 }st,stclass[30], * p;
```

对结构成员的引用，涉及运算符·和→，对结构类型变量或数组元素的成员用运算符·，而对指针所指结构类型变量或数组元素的成员用运算符→。如

```
st.num          p→num
```

st.name　　p→name

st.add.post　　p→add.post

注意，此时两边并不等价，只有当

p=&st

时，即 p 指向结构变量 st，上述两边才是等价的。

三、联合

联合类型说明的一般形式为

```
union  联合名{
  数据类型  成员名1;
  数据类型  成员名2;
    …
  数据类型  成员名 n;
};
```

联合类型的定义类似于结构类型，本质不同是不同数据类型的数据共用存储区域。联合类型所需存储单元字节数为联合成员中所占字节的最大值。联合类型中成员的引用方式类似于结构，也是通过运算符·和→实现的，但联合在每一时刻只能引用其中一个成员，且这个成员必须是最近存储的那个。

四、枚举

一般形式

```
enum  枚举名{枚举表};
```

其中枚举表是一个有限集合，均为标识符，说明往后枚举变量的可能取值只能是这些标识符之一。说明了枚举类型后，可以类似于结构、联合一样，定义枚举变量。如

```
enum day{
  Sun,Mon,Tue,Wed,Thu,Fri,Sat};
enum day d;
```

定义 d 为枚举型 enum day 的枚举变量，取值只能为 Sun、Mon、…、Sat 之一。

编译程序实际上把枚举值按常量处理，即枚举常量与整数相联系。从枚举表中的第一枚举元素起，以0开始把连续整数“赋给”这些枚举元素，即 Sun 的值为0，Mon 的值为1，…，Sat 的值为6。这种隐含的表示可以通过在定义时向某个枚举元素“赋”常量来加以改变，其后的枚举元素值就从这个常量开始递增。

第四节　类 型 定 义

类型名有系统提供的 char、int、float 和 double 等；用户定义的结构、联合和枚举类型等。此外，还可以通过类型定义 typedef 语句引入新的类型名，typedef 语句允许用户为已有数据类型名引入新的类型名。引入后两者同时有效。

一般形式为

typedef　类型名　新名称；

如　　typedef　int　INTEGER；

则 INTEGER 等价于 int。

```
typedef struct{
  long num;
  char name[20];
  float score;
}    STUDENT;
```

则 STUDENT 等价于前面的结构类型。

typedef 语句除了可以对基本类型、结构类型等引入新的类型名外，还可以在 typedef 语句中加入符号 *、[整型常量表达式]等，使引入的类型名是指针型、数组型和结构数组型。如

```
typedef char *STRING;
typedef double SCORE[30];
```

则 STRING 等价于字符指针型，SCORE 等价于 double 型数组[30]。经过类型定义后，如下的定义语句

```
INTEGER i,j;
STRING p;
SCORE a;
STUDENT stclass[30];
```

的含义分别是

i,j 为整型变量；

p 为字符型指针变量；

a 为 double 型数组，有30个元素；

stclass 为结构型的结构数组，有30个元素。

从上看出，经过 typedef 定义的新名称本身可以含有指针、数组内容。

第五节　文 件 类 型

文件是数据的有序集合。操作系统把设备也称为文件，所以文件是磁盘文件、光盘文件、磁带文件等和有 I/O 能力的外设的总称。

在 C 语言中，把数据流看作是对文件的一次抽象。文件可以是各不相同而流是统一的，使用流可以实现对文件的统一处理。

ANSI C 描述两个类型的流：

(1)文本流：是组织成行的字符序列，每行是由0个或多个字符组成，以换行符'\n'结尾(不是必需的)，文本流是以字符为单位。

在有的编译实现时，文本文件中字符'\r'、'\n'序列，映射到文本流中为一个字符'\n'。

(2)二进制流：是一个简单字节序列。它是以数据在内存中的存储形式(二进制码)组成的字节序列。文件与二进制流之间的数据是一一对应的，但在序列的末尾可能含附加一个或多个空格。