第2章　基本数据类型及输入输出

2.1 C语言的数据类型

数据是算法处理的对象，包括数值数据、文字数据、图像数据、声音数据等。C语言中的数据以某种特定的形式存在，如整型、实型、字符型等。

C语言中的数据类型包括基本数据类型、构造类型、指针类型和空类型。

基本数据类型简称基类型。主要特点是其值不可以再分解为其他类型。有整型、实型、字符型3种（枚举型已淘汰）基本数据类型。

构造数据类型是根据已定以的一个或多个数据类型用构造的方法组合而成的一种数据类型。

指针类型是一种特殊的、具有重要作用的数据类型，其值用来标识某个量在内存中的地址。

数据类型确定了数据在内存中所占存储空间的大小，即确定了其表示的范围，如表2.1（书）。

[Return]

2.2 标识符、常量和变量

2.2.1 标识符

在C语言中，标识符是程序的基本语法单位。程序设计过程中用到的各类元素的名字都叫标识符。标识符可作为变量名、符号名、函数名和数组名、文件名等。正确的使用标识符和关键字对编制程序至关重要。C语言的标识符可以分为以下三类：

一、关键字

所谓关键字就是已被Turbo C2.0本身使用，不能作其它用途使用的字。关键字不能用作变量名、函数名等。 如：int,float,if等都是关键字，关键字都为小写字母。

(1)数据类型关键字，包括char,double,enum,float, int,long,short,signed,struct,union,unsigned,void共12个。

(2)控制语句关键字，包括break,case,continue,default,do,else,for,goto,if,return,switch,while共12个。

(3)存储类别关键字，包括auto,extern,register,static共4个。

(4)其他关键字，包括const,sizeof,typedef,volatile共4个。

二、预定义标识符

在C语言中也都有特定含义，如库函数名（printf）和预编译处理命令（define）等。C语言语法允许这类标识符另作它用，但原有含义将丢失。为此，建议不要把这类标识符另作它用。

三、用户标识符

由用户根据需要定义的标识符成称为用户标识符。一般用来给变量、函数、数组或文件等命名。标识符是用于标识某个量的符号，因此，命名应尽量有相应的意义，以便阅读理解，作到“顾名思义”。 用户标识符在使用时应避免和关键字及预定义标识符相同，否则将出现错误。

用户标识符是一个名称，不同的程序设计语言使用不同形式的标识符。在C语言中用户标识符的命名规则如下：

（1）有效字符：只能由字母、数字和下划线组成，且以字母或下划线开头。

（2）有效长度：随系统而异，但至少前 8 个字符有效。如果超长，则超长部分被舍弃。

例如，由于student_name和student_number的前 8 个字符相同，有的系统认为这两个变量，是一回事而不加区别。

（3）大小写字母表示不同意义, 即代表不同的标识符。习惯上，变量名和函数名中的英文字母用小写，以增加可读性。

[Return]

2.2.2 常量和符号常量

1.常量

在程序运行过程中，其值不能被改变的量称为常量。常量可分为4类：整型常量、实型常量、字符常量和字符串常量。

2.符号常量

可以使用一个标识符表示常量，但符号名必须在程序中先定义，然后才能使用。其一般形式为：

```
#define 标识符 常量
```

其中#define也是一条预处理命令（预处理命令都以“#”开头），称为宏定义命令。其功能是把该标识符定义为其后的常量值。一经定义，以后在程序中所有出现该标识符的地方均代之以该常量值。习惯上符号常量的标识符用大写字母，变量标识符用小写字母，以示区别。符号常量不是变量，它所代表的值在整个作用域内不能再改变。也就是说，在程序中，不能再用赋值语句对它重新赋值。

例2.1（P17）

[Return]

2.2.3 变量

在程序运行过程中，其值可以被改变的量称为变量。变量的两个要素：

（1）变量名。每个变量都必须有一个名字——变量名，变量命名遵循标识符命名规则。

（2）变量值。在程序运行过程中，它在内存中占据一定的存储单元，该存储单元用来存放变量的值。在程序中，通过变量名来引用变量的值。允许多次给变量赋值，但新值会覆盖原值。

1.变量名

变量名属于用户标识符。

2.变量的定义

在C语言中，要求对所有的变量先定义，后使用。说明就是给变量命名。变量定义的一般格式为：

数据类型关键字　变量1，变量2，…，变量n;

在C语言中使用变量前必须先定义，原因为：

(1)凡未被事先说明的变量，不能作为变量名。

(2)在编译时系统自动为每个变量分配相应的存储单元，因此必须在定义时为每个变量指定一个确定的类型。

(3)为了便于检查变量在编译时所进行的运算是否合法，规定每个变量只能属于一种类型。

在书写变量说明时，应注意以下几点：

（1）允许在一个类型说明符后，说明多个相同类型的变量。各变量名之间用逗号间隔。类型说明符与变量名之间至少用一个空格间隔。

（2）最后一个变量名之后必须以“；”号结尾。

（3）变量说明必须放在变量使用之前。一般放在函数体的开头部分。

3.变量的初始化

可以在定义变量的同时给变量赋初值，也称变量初始化。

如： `int i=1,j=0,k=2;`

`int i=1, j=1;`

以下定义错误： `int i=j=1;`

但可以：`int i，j;`

`i=j=1;`

[Return]

2.3 整型数据

2.3.1 整型常量

整型常量即整常数，在C语言中可用三种形式表示：

（1）十进制。例如10、36。

（2）八进制（以数字 0 开头）。数码取值为0～7。八进制数通常是无符号数 。例如012。

（3）十六进制（以数字 0 +小写字母x或大写字母X开头）。其数码取值为0~9，A~F或a~f。例如0x36。

2.3.2 整型变量

1.变量类型

根据占用内存字节数的不同，整型变量又分为 4 类：

（1）基本整型（类型关键字为[signed] int）。

（2）长整型（类型关键字为[signed] long [int]）。

（3）无符号基本整型。（类型关键字为unsigned [int]）

（4）无符号长整型（类型关键字为unsigned long [int]）

无符号型只能用来存储大于等于零的整数。若不指定为无符号型，隐含的即为有符号型（signed)。有符号型可以用来存储正、负整数。

2.整型变量的长度和取值范围

上述各类型整型变量占用的内存字节数，随系统而异。在16位操作系统中，一般用 2 字节表示一个int型变量，且long型（ 4 字节）≥int型（ 2 字节）≥short型（ 2 字节）。显然，不同类型的整型变量，其值域不同。占用内存字节数为n的（有符号）整型变量，其值域为：-2^{n*8-1}~（$2^{n*8-1}-1$）；无符号整型变量的值域为：0~（2^{n*8-1}）。

例如，PC机中的一个int型变量，其值域为-2^{2*8-1}~（$2^{2*8-1}-1$），即-32768~32767；一个unsigned型变量的值域为：0~（2^{2*8-1}），即0~65535。

3.整型变量的定义

数据类型关键字 变量1，变量2，…，变量n;

例如： int a,b,c; (a,b,c为整型变量)

long x,y; (x,y为长整型变量)

unsigned p,q; (p,q为无符号整型变量)

4.整数在内存中的存储形式

通常将一个字节中最右边的一位称为最低位，最左边的一位称为最高位。在C语言中，一个int整数用两个字节存放，最高位用来存放整数的符号，如果是正整数，最高位置0，如果是负整数，最高位置1。

一、正整数（以int整数为例）

C语言中正整数以原码形式存储，即将该整数转换为二进制代码形式存放。对于正整数,它的原码、反码、补码相同。例如：正整数97在内存中的存储形式为：0000000000100001。由此可见，所能存放的最大正整数为：0111111111111111，十进制形式为：32767。

二、负整数

1）C语言中负整数以补码形式存储，负整数的原码、反码、补码不同。举例说明何谓补码，以-5为例。

（1）求二进制形式，得出1000000000000101；

（2）对其余各位求反，得出反码形式。即：把1转换为0，把0转换为1。得出1111111111111010。

（3）求补码。即：在反码的基础上加1。得出1111111111111011。

2）如何将以补码形式存放的二进制代码转换成十进制的负整数，使用的方法是以上操作的逆操作。

（1）对各位取反。例如：有补码1111111111111011，取反后为1000000000000100。

（2）转换为十进制。例如：0000000000000100的十进制数为4。

（3）在十进制数前加负号，得-4。

（4）对该数再减1，得-5。

由此可见，所能存放的最大负整数为：1111111111111111，十进制形式为：-1；所能存放的最小负整数为：1000000000000000，十进制形式为：-32768。

三、无符号整数

无符号整数在存储时，因为没有负整数，所以二进制位的最高一位不是符号位，而是用来存放数据。由此可见，所能存放的最大正整数的二进制形式为：1111111111111111，十进制形式为：65535。

5.整型数据的溢出

当数据超出默认范围时就会发生溢出现象。溢出时系统并不报错，但看不到正确结果。

[Return]

2.4 实型数据

2.4.1 实型常量

实型常量即实数，在C语言中又称浮点数，其值有两种表达形式:

(1)十进制小数形式。例如3.14、9.8、.45、36.。

(2)指数形式：<尾数>E（e）<整型指数>。例如3.0E+5等。中间不能有空格。

注意：

（1）E（e）的前端必须有数字，E（e）的后端必须是整数。

（2）规范化的指数形式为E（e）前端的数字的整数部分必须是一位非零整数。

2.4.2 实型变量

实型变量分为3类：单精度型、双精度型和长双精度型。

（1）单精度型。类型关键字为float，一般占 4 字节、提供 7 位有效数字。

（2）双精度型。类型关键字为double，一般占 8 个字节、提供15~16位有效数字。

（3）长双精度型。类型关键字为long double，一般为10个字节、提供18～19位有效数字。一般不用该类型。

有效位是指数据在几位之内为有效数字。实数在存储时数值都先化成指数，再用二进制存储。在计算机内存中可以精确存放一个整数，不存在误差，数值范围小；实型数在存储时都有误差，数值范围大。

例：书P21-22

[Return]

2.5 字符型数据

2.5.1字符型常量

字符型常量包括：字符常量、字符串常量。

字符常量是用单引号括起来的一个字符，代表ASCII字符集中的一个字符。例如：‘a’,‘b’,‘=’,‘+’,‘?’都是合法字符常量。

在C语言中，字符常量有以下特点：

1)字符常量只能用单引号括起来，不能用双引号或其它括号。

2)字符常量只能是单个字符，不能是字符串。

3)字符可以是字符集中任意字符。但数字被定义为字符型之后就和原数值不等价。如’5’和5 是不同的。’5’是字符常量。

字符在计算机中存放的是ASCII代码值，在内存中占1个字节。 有符号字符型数取值范围为-128-127, 无符号字符型数到值范围是0-255。因此在 C语言中, 字符型数据在操作时将按整型数处理。在计算机内部，一个字符常量就是ASCII码字符集中该字符的序号。因此字符常量‘A’的值为65（八进制101），‘a’的值为97（八进制141），‘0’的值为48（八进制60），空格字符的值为32（八进制40）。

2. 5. 2转义字符常量(反斜杠字符常量)

转义字符是一种特殊的字符常量。转义字符以反斜线“\”开头，后跟一个或几个字符。转义字符具有特定的含义，不同于字符原有的意义，故称“转义”字符。例如，在前面各例题printf函数的格式串中用到的“\n”就是一个转义字符，其意义是“回车换行”。转义字符主要用来表示那些用一般字符不便于表示的控制代码。

常用的转义字符及其含义如下：

\n　回车换行　　\t　横向跳到下一制表位置
\v　竖向跳格　　\b　退格
\r　回车,移到本行开头　\\　反斜线符“\”
\’　单引号符　　\0　空值
\ddd　1～3位八进制数所代表的字符
\xhh　1～2位十六进制数所代表的字符

广义地讲，C语言字符集中的任何一个字符均可用转义字符来表示。\ddd和\xhh正是为此而提出的。ddd和hh分别为八进制和十六进制的ASCII代码。如‘\101’表示字符‘A’，‘\102’表示字母‘B’，‘\134’表示反斜线，‘\xOA’表示换行等。

2.5.3字符串常量

严格的说，这不是一种数据类型。只有常量，没有变量。一个字符串常量是用一对双引号括起来的一串字符。如：“Hello,World!\n”就是字符串常量。在printf和scanf中的格式控制部分是由一对双引号括起来的一串字符，因此也是字符串常量。在C语言中，系统在每个字符串的最后自动加入一个字符-’\0’作为字符串的结束标志。

字符串在内存中存储时所占的字节数不定，为字符的个数加1。即每一个字符占一个字节，最后一个字节存放'\0'。

例如，“How do you do.”、“Good morning.”等，都是字符串常量，其长度分别为14和13（空格也是一个字符）。

如果反斜杠和双引号作为字符串中的有效字符，则必须使用转义字符。

例如：（1）C:\msdos\v6.22 → "C:\\msdos\\v6.22"

（2）I say:"Goodbye!" → "I say:\"Goodbye!\""“

注意：在源程序中书写字符串常量时，不必加结束字符‘\0’，否则画蛇添足。

如果有一个字符串为“CHINA”，则它在内存中的实际存储如下所示：

最后一个字符'\0'是系统自动加上的，它占用 6 字节而非 5 字节内存空间。

C	H	I	N	A	\0

2.5.4可对字符量进行的运算

在C程序中，字符量可参与任何整数运算。例如：

’B’-’A’≡66-65≡1

此处≡表示“等价”关系。其中66，65都是十进制数，分别是B、A的ASCII值。很容易利用算术运算把大写字母转换成小写字母或把小写字母转换成大写字母。

也可以通过算术运算将一个数字字符转换成整数或把一位整数转换成数字字符。

在C语言中，字符量也可以进行关系运算和逻辑运算。如：'a'<'c'，由于在ASCII码表中，'a'的值小于'c'的值，因此这个关系运算的结果为“真”。如果进行逻辑运算，如：'a'&&'c'，由于在ASCII码表中，'a'、'c'的值都为非零值，因此“与”运算的结果为1。

2.5.5 字符型变量

字符型变量的类型关键字为char，一般占用1字节内存单元。

1.字符型变量的定义

字符型变量用来存储字符常量。将一个字符常量存储到一个字符变量中，实际上是将该字符的ASCII码值（无符号整数）存储到内存单元中。

例如，

```
char ch1, ch2;
ch1='a'; ch2='b';
```

2.字符型变量与整型变量的关系

字符数据在内存中存储的是字符的ASCII码—— 一个无符号整数，其形式与整数的存储形式一样，所以C语言允许字符型数据与整型数据之间通用。

（1）一个字符型数据，既可以字符形式输出，也可以整数形式输出。

例1: 字符型变量的字符形式输出和整数形式输出。

```
main()
{ char ch1,ch2;
  ch1='a'; ch2='b';
  printf("ch1=%c,ch2=%c\n",ch1,ch2);
  printf("ch1=%d,ch2=%d\n",ch1,ch2);        }
```

程序运行结果：

ch1=a,ch2=b

ch1=97,ch2=98

（2）允许对字符数据进行算术运算，此时就是对它们的ASCII码值进行算术运算。

例2: 字符数据的算术运算。

```
main()
{ char ch1,ch2;
  ch1='a'; ch2='B';
  printf("ch1=%c,ch2=%c\n",ch1-32,ch2+32);
```

```
        printf("ch1+200=%d\n", ch1+200);
         printf("ch1+200=%c\n", ch1+200);
         printf("ch1+256=%d\n", ch1+256);
         printf("ch1+256=%c\n", ch1+256);     }
```

程序运行结果：

```
ch1=A,ch2=b
ch1+200=297
ch1+200=)
ch1+256=353
ch1+256=a
```

例：书P23-24

思考题：用字符形式输出一个大于256的数值，会得到什么结果？

[Return]

2.6 各类数值数据间的混合运算

三大基本数据类型间可以进行混合运算。要实现混合运算，就必须进行类型转换。

在进行运算时，不同类型的数据要先转换成同一类型，然后进行运算。转换的基本原则是将整型转换为实型。

int(short,char)—>unsigned—>long—>double(float)

(2)在C语言中，所有实型数的运算均以双精度方式进行。若是单精度数，则在尾数部分补0，使之转化为双精度数。

注意：以上介绍的是一般算数转换，这种类型转换由系统自动进行。

[Return]

2.7 字符输入输出函数

2.7.1 putchar函数

1．putchar()函数的格式：

putchar(ch);

其中ch可以是一个字符变量或常量，也可以是一个转义字符，还可以是整型变量或常量。

2．putchar()函数的作用：向终端输出一个字符。

（1）putchar()函数只能用于单个字符的输出，且一次只能输出一个字符。另外，从功能角度来看，printf()函数可以完全代替putchar()函数。

（2）在程序中使用putchar()函数，务必牢记：在程序（或文件）的开头加上编译预处理命令（也称包含命令），即：

#include "stdio.h"

表示要使用的函数，包含在标准输入输出（stdio）头文件（.h）中。

例：书P59-60

例1: putchar() 函数的格式和使用方法。

```
#include "stdio.h"
 main()
{
  char ch1='N', ch2='E', ch3='W';
  putchar(ch1); putchar(ch2); putchar(ch3);
  putchar('\n');
  putchar(ch1); putchar('\n');
  putchar('E'); putchar('\n');
  putchar(ch3); putchar('\n');
}
```

程序运行结果如下：

```
NEW
N
E
W
```

2.7.2 **getchar()**函数（字符输入函数）

1.getchar()函数的格式：**getchar();**

2.getchar()函数的作用：从系统隐含的输入设备（如键盘）输入一个字符。另外，从功能角度来看，**scanf()**函数可以完全代替**getchar()**函数。

（**1**）**getchar()**函数只能用于单个字符的输入，一次输入一个字符。

（**2**）程序中要使用**getchar()**函数，必须在程序（或文件）的开头加上编译预处理命令：

#include "stdio.h“

例: 说明**getchar()**函数的格式和作用。

```
#include "stdio.h"
main()
{char  ch;
  printf("Please input two character: ");
  ch=getchar();
```

```
 putchar(ch);
 putchar('\n');
 putchar(getchar());
 putchar('\n');
 }
```

程序运行情况如下：

```
Please input two characters: ab↙
a
b
```

例：书**P63-64**

[Return]

2.8 格式输入输出函数

把数据从计算机内部送到计算机的外部设备上的操作称为“输出”。C语言没有专门的输出语句，必须通过调用输出库函数来完成。 **printf()函数的作用：向计算机系统默认的输出设备（一般指终端或显示器）输出一个或多个任意类型的数据。**

2.8.1 printf()函数（格式输出函数）

printf()函数是标准输出函数，一般用于向标准输出设备按规定格式输出信息。在编写程序时经常会用到此函数。printf()函数的调用格式为:

printf(″格式控制″, 输出项表)

在该函数的末尾添“；”，就构成了输出语句。

例如：printf(“a=%d,b=%f,c=%e”,a, b, c);

1．格式控制。是由一组“ ”引起来的一串字符串，也称“格式字符串”或“转换控制字符串”，可以包含三种信息：

（1）格式说明。格式说明的一般形式如下：

%[标志][宽度][.精度][h|L]格式说明符

即以“%”开始, 以一个规定字符结尾,这个字符被称为格式说明符（类型转换字符），用来确定输出内容的格式。例如：当输出项是int类型时，系统规定用d作为格式说明符，其形式为%d；当输出项是float或double类型时，系统规定用f或e作为格式说明符，其形式为%f或%e（对于double类型也可用%lf或%le）。

（2）转义字符

例如，'\n'就是转义字符，输出时产生一个“换行”操作。

（3）普通字符——除格式说明和转义字符之外的其它字符。格式字符串中的普通字符，将原样输出。

例如，“printf(“radius=%f\n”, radius);”语句中的“radius=”，

“printf(“length=%7.2f,area=%7.2f\n”, length,area);”语句中的“length=”、“area=”等都是普通字符。

2．输出项表

输出项表中列出的是需要输出的一系列参数,输出项可以是常量、变量或表达式。其个数必须与格式控制中格式说明的个数一样多,类型也应该相匹配，且顺序一一对应,否则将会出现意想不到的错误。输出项表是可选的。如果要输出的数据不止1个，相邻两个之间用逗号分开。下面的printf()函数都是合法的：

（1）printf("I am a student.\n");

（2）printf("%d",3+2);

（3）printf("a=%f　　b=%5f\n", a, a+3);

3. printf函数中常用的格式说明

格式说明总是以“%”开始，以一个格式说明符结尾,在此之间可以根据需要插入“宽度说明”、左对齐符号“-”、前导零符号“0”等。 输出不同类型的数据，要使用不同的格式说明符。

1）格式说明符

(1) 类型转换字符d——以带符号的十进制整数形式输出。

例1 类型转换字符d的使用。

```
int  num1=123;
printf("num1=%d”， num1);
```

对于整数，还可用八进制无符号形式（%o(小写字母o)）和十六进制无符号形式（%x）输出。对于unsigned型数据，也可用%u格式符，以十进制无符号形式输出。

所谓无符号形式是指，不论正数还是负数，系统一律当作无符号整数来输出。

（2）类型转换字符c——以字符形式输出字符型数据。

（3）类型转换字符f——以小数形式、按系统默认的宽度（六位小数），输出单精度和双精度实数。

对于实数，也可使用格式符%e，以规范指数形式输出：尾数中的整数部分大于等于1、小于10，小数点占一位，尾数中的小数部分占5位；指数部分占4位（如e-03），其中e占一位，指数符号占一位，指数占2位，共计11位。

也可使用格式符%g，让系统根据数值的大小，自动选择%f或%e格式、且不输出无意义的零。

2）长度修饰符

长度修饰符加在"%"和格式字符之间，对于长整型数必须加“l”，双精度实型数可以加也可以不加“l”。

3）输出数据所占的宽度

如果没有特别声明，输出数据的宽度由系统默认决定，并采用右对齐的形式。可以在"%"和格式字符之间插进数字表示数据输出的最大宽度。

（1）在“%”和格式字符之间插入一个整数来指定数据的输出宽度，采用“m”形式。如果数据的实际位数超过指定的宽度，将按其实际长度输出；如果数据的实际位数小于指定的宽度，数据右对齐，左边补空格。

（2）对于**float**或**double**型数据可以用“**m.n**”形式来指定数据的输出宽度。其中**m**表示输出数据的总宽度，**n**表示输出数据的小数位数，称为精度。这种形式优先考虑**n**。

若小数部分位数小于指定的小数位宽度,则在末尾补0；若小数部分位数超过了指定的小数位宽度,则按定义的宽度以“四舍五入”方式输出。若输出数据所占的宽度小于指定的总宽度,在左边补空格；若输出数据所占的宽度超出指定的总宽度，将先满足小数位数，整数部分原样显示。

（3）实型数据用“.n”形式定义。表示指定输出数据的小数位数，整数部分原样显示。如果n为零，表示所有小数位数（连同小数点）一起被隐藏。

注意：输出数据的实际精度并不取决于输出当中所定义的宽度和小数位数，而是取决于数据在计算机内的存储精度。也就是说，利用输出语句对数据进行输出时，数据的实际大小不会改变，只会以你所希望的形式显示一个数。

4）输出数据左对齐

可以在宽度前加一个“-”来实现输出数据的左对齐。

5）使输出的数字总是带有+号或-号

可以在宽度前加一个“+”来实现该功能。

6）在输出数据前加前导0

若想在输出值前加一些0，就应在宽度项前加个0。

7）在输出的八进制数前加前导0，在输出的十六进制数前加前导0x。

用格式说明符o或x输出八进制数和十六进制数时，输出的数据前并不出现相应的前导0和0x，如果希望前导出现，可在%和格式字符间插入#号。

例如：printf(“%-6d,%06d,%-06d”,123,12,12);
例如：printf(“%d,%+d”,45,38);
输出结果依次是:**45,+38**。
例如：printf(“%o,%#o,%x,%#x”,10,10,10,10);
输出结果依次是：12,012,a,0xa。
例如：printf(“%9f,%12f”,123.4587,123.4587);
例如：printf(“%3d,%9.2f”,24,123.4587);
例如：printf(“%3d,%6.2f”,2417,1235.4537);
输出结果依次是：2417,1235.45。
例如：printf(“%.2f,%.0f”,1235.4537,1235.4537);
例如：printf(“%-14.8f,%014.5f”,1.3455,3.1415);

4. 调用printf函数时的注意事项：

（1）格式控制中的格式说明符，必须按从左到右的顺序，与输出项表中的每个数据一一对应(类型上以及数量上)，否则出错。在输出长整型数据时，必须在%与格式字符间加l。

（2）格式字符x、e、g可以用小写字母，也可以用大写字母。使用大写字母时，输出数据中包含的字母也大写。除了x、e、g格式字符外，其它格式字符必须用小写字母。

（3）格式字符紧跟在“%”后面就作为格式字符，否则将作为普通字符使用（原样输出）。

（4）要想输出一个%，在格式控制中必须连续输入%%。 例如：

```
printf(“%%f”,1.0/3);
```

2.8.2 scanf函数（格式输入函数）

scanf()函数是用来从外部输入设备向计算机主机输入数据的。在C语言中，数据的输入同样是借助库函数来完成的，scanf函数是标准输入库函数 。

在程序中给计算机提供数据，可以用赋值语句，也可以用输入函数。在C语言中，可使用scanf()函数，通过键盘输入，给计算机同时提供多个、任意的数据。

1 . scanf函数的一般调用形式

其调用格式为:

scanf(格式控制, 输入项表)

在函数的末尾加上“；”，就构成了输入语句。

（1）格式控制（格式控制串）。格式控制串可以包含2种类型的字符：格式说明（格式转换说明）和其他字符（又称普通字符）。

格式说明与printf()函数的相似，用来指定输入时的数据转换格式，普通字符必须原样一起输入。

（2）输入项表（输入项首地址表）——由若干个输入项首地址组成，各输入项必须是地址表达式,相邻2个输入项首地址之间，用逗号分开。

输入项首地址表中的地址，是变量的首地址。

变量首地址的表示方法：　&变量名

其中“&”是地址运算符。

例：　scanf("%d, %f", &i, &j);

2. scanf函数中常用的格式说明

格式说明的一般形式为：

% [*] [宽度] [h|l] 格式字符

格式说明必须以%开头，以一个格式字符结尾。其格式字符如下表所示。

d	输入十进制整数	o	输入八进制整数
x	输入十六进制整数	u	输入无符号十进制整数

f或e 输入实型数(用小数形式或指数形式)

c 输入单个字符 s 输入字符串

说明：

（1）在输入长整型数据时，必须在%与格式字符间加l；在输入双精度实型数时，也必须在%与格式字符间加l。

（2）scanf函数中没有精度控制，如： scanf("%5.2f",&a);是非法的。即不能对实型数指定小数位数。但可以对数据定义输入数据的总宽度。

（3）格式说明与输入项从左到右在类型上以及数量上都必须匹配。若类型不匹配，将不能正确接收数据；若数量不匹配，当格式说明的个数少于输入项的个数时，多余的数据不被接收，当格式说明的个数多于输入项的个数时，scanf函数提前结束。如：

scanf(“%d%d”,&a,&b,&c);

scanf(“%d%d%d”,&a,&b);

都只能正确接收a,b的值，c值无法接收。

3. 通过scanf函数从键盘输入数据

1）输入数值数据

当调用scanf函数从键盘输入数据时，必须以回车键（Enter键）作为结束，这样该函数才能接受从键盘输入的数据。

从键盘输入数值数据时，相邻的2个数据之间，至少用一个空格，TAB或回车符作间隔符，数量不限。

例如，scanf("%d%d",&num1,&num2);

假设给num1输入12，给num2输入36，则正确的输入操作为：

12□36↙　　或者：12↙

36↙

注：“↙”符号表示按回车键操作，通知系统输入操作结束。

2）指定输入数据所占宽度m

读取输入数据中相应的m位，但按需要的位数赋给相应的变量，多余部分被舍弃。当输入数据时，C系统并不要求必须

按指定的宽度输入数据，可以像未指定宽度时输入。若一次输入多个数值数据，并且没有输入默认间隔符，可利用这种方式截取数据。

例如：scanf(″%5d″,&a);

输入：12345678只把12345赋予变量a，其余部分被截去。

又如： scanf(″%4d%4d″,&a,&b);

输入：12345678，将把1234赋予a，而把5678赋予b。

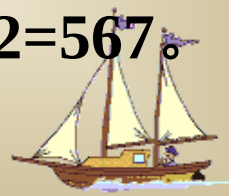
3）跳过输入数据的方法

可在格式字符和“%”间加一个“*”号，表示本输入项对应的数据读入后，不赋给相应的变量（该变量由下一个格式说明输入）。

例如，scanf("%2d%*2d%3d",&num1,&num2);

printf("num1=%d,num2=%d\n",num1,num2);

假设输入“123456789”，则系统将读取“12”并赋值给num1；读取“34”、但舍弃掉（“*”的作用）；读取“567”并赋值给num2。所以，printf()函数的输出结果为：num1=12,num2=567。



4）输入的数据少于 scanf函数要求输入的数据

scanf函数将等待输入，直到满足要求或遇到非法字符为止。

5）输入的数据多于 scanf函数要求输入的数据

多余的数据将留在缓冲区作为下一次输入操作的输入数据。

6）在格式控制串中使用普通字符

“格式字符串”中出现的普通字符（包括转义字符形式的字符），务必原样输入。

例如，scanf("%d,%d",&num1,&num2);

假设给num1输入12，给num2输入36，正确的输入操作为：12，36↙

另外，scanf()函数中、格式字符串内的转义字符(如\n)，系统并不把它当转义字符来解释，从而产生一个控制操作，而是将其视为普通字符，所以也要原样输入。

例如：scanf("num1=%d,num2=%d\n",&num1,&num2);

假设给num1输入12，给num2输入36，正确的输入操作为：

num1=12，num2=36\n↙

提高人机交互性建议：为改善人机交互性，同时简化输入操作，在设计输入操作时，一般先用printf()函数输出一个提示信息，再用scanf()函数进行数据输入。

例如，将scanf("num1=%d,num2=%d\n",&num1,&num2);改为：

```
printf("num1="); scanf("%d",&num1);
printf("num2="); scanf("%d",&num2);
```

7）类型修饰符——h、l（了解）

其含义与printf()中的一样，分别为短整型和长整型。

注意:输入数据时，遇到以下情况，系统认为该数据结束：

（1）遇到空格，或者回车键，或者Tab键。

（2）遇到输入域宽度结束。例如“%3d”，只取3列。

（3）遇到非法输入。例如，在输入数值数据时，遇到字母等非数值符号(数值符号仅由数字字符0-9、小数点和正负号构成)。

4.用scanf函数输入字符

要使用“%c”作为格式说明符。如：

```
char a,b;
scanf(“%c%c”,&a,&b);
```

注意：当连续几个格式说明符“%c”紧凑书写时，在输入字符中字符间没有间隔符。因为间隔符空格、回车以及横向跳格符都将作为一个字符被读入。例如执行上面语句时，若输入x y，则在变量a中存放字符'x'，在变量b中存放字符' '，将字符'y'留在内存中供下一次使用。也可以在格式控制串中加空格，如：

```
scanf(“%c    %c”,&a,&b);
```

输入的形式可以和不加空格的scanf相同，但这时间隔符-空格、回车以及横向跳格符都将作为分隔符使用，不能代表字符输入。若要给变量 a,b分别输入字符'x'、'y'，可用以下几种形式： xy或x y

可以指定输入字符宽度，此时在输入数据时应严格按指定的宽度输入数据，并且取输入宽度中的第一个字符作为输入的数据。如要给变量 a,b分别输入字符‘x’、‘y’。

```
char a,b;
scanf("%3c%3c",&a,&b);
```

正确的输入形式是:x y ，而不是: x y。

当交叉输入数值数据和字符数据时，要严格按照规定输入，以免出错。如：

```
int a,b;    char c,d;
scanf("%d%c%d%c",&a,&c,&b,&d);
```

应输入：10x20y，或: 10x 20y。若输入： 10 x 20 y，将把10给a,把' '给c，马上终止scanf语句，b,d没有接收数据。因此，字符-数值间可以加间隔符，数值-字符间不能加间隔符。

练习题：

1.正确的整型常量是

A.12.　　B.-20　　C.1,000　　D.4 5 6

2.正确的实型常量是

A. 0　　B.3. 1415　　C.0.329×10^2　　D. .871

3.不正确的实型常量是

A. 2.607E-1　　B.0.8103E 2　　C.-77.77　　D. 456E-2

4.不合法的用户标识符是

A. abc.c　　B.file　　C.Main　　D. PRINTF

5.不合法的用户标识符是

A. _123　　B.printf　　C.A!　　D. Dim

6.合法的用户标识符是

A. int define WORD　　B.as_b3 _123 If

C. For -abc case　　D. 2c DO SIG

7.不合法的八进制数是

A. 0　　B.028　　C. 077　　D. 01

8.以下不能定义为用户标识符的是

A.Main　　B._0　　C. if　　D._abc

9.以下选项中，属于字符串常量的是

A.How are you　B.'abcd'　C."chinese"　D.MYMabcd

10.字符串"\\\22a,0\n"的长度是

11.不是合法的关键字的是

A.long　　B.double　　C.Int　　D.signed

12.以下能正确地定义整型变量a,b和c并为其赋初值5的语句是

A. int a=b=c=5;　　B. int a,b,c=5;

C. int a=5,b=5,c=5;　　D. a=b=c=5;

13.如果x为 float型变量，则以下语句输出为

x=213.82631;　printf("%4.2f\n",x);

A.宽度不够，不能输出　B.213.82　C.213.82631　D.213.83

14.若从键盘输入9876543210，下列程序的输出结果是

```
main()
{ int a;   float b,c;
  scanf(“%2d%3f%4f”,&a,&b,&c);
   printf(“a=%d,b=%f,c=%f\n”,a,b,c);   }
```

A.a=98,b=765,c=4321　　B. a=10,b=432,c=8765

C. a=98,b=765.000000,c=4321.000000

D. a=98,b=765.0,c=4321.0

15.下列语句的输出结果是

```
   long  a=0xffffffff;     int  b=a;
   printf(“%d”,b);
```

A.65535　　B.65536　　C.-1　　D.1

16.’\101’常量是

A.字符A　　B.字符a　　C.字符e　　D.非法的常量

17. 若有定义int a=2,b=5;以下程序段的输出结果是：

printf(“a=%%d,b=%%d\n”,a,b);

A.a=%2,b=%5　　B.a=2,b=5

C.a=%%d,b=%%d　　D.a=%d,b=%d

18.若x=496,以下程序段的输出是：

printf(“*%-06d*\n”,x);

A.*496　*　　B.*　496*　　C.*000496*　　D.*496000*

19.若有int x=0177;以下程序段的输出结果是_____：

printf(“%-2d,%-6d,$%-06d,$%06d,%%06d”,x,x,x,x,x);

20.若有int x=0177;以下程序段的输出结果是_____：

printf(“%2d,%6d,%6o,%#6o,%6x,%#6x,%6u”,x,x,x,x,x,x,x);

21.若有double x=513.789215，以下程序段的输出结果是_____：

printf(“%8.6f,%8.2f,%14.8f,%6.3lf\n”,x,x,x,x);

22.以下程序段的输出是_：

A.|3.1415|　　B.|　3.0|　　C.|　　3|　　D.|　 3.|

float　a=3.1415;　 printf(“|%6.0f|\n”,a);

23.以下程序段的输出是_：

A.|2345.67800|　　B.|12345.6780|

C.| 12345.67800|　　D.| 12345.678|

printf(“|%10.5f|\n”,12345.678);

24.以下程序段的输出是_：

A.*0000057.66*　　B.*　 57.66*

C.*0000057.67*　　D.*　 57.67*

float　a=57.666;　 printf(“*%010.2f*\n”,a);

[Return]